03-07-02

A

ASSISTANT COMMISSIONER FOR [redacted]NTS
BOX PATENT APPLICATION
WASHINGTON , D.C. 20231

CASE DOCKET NO: JP[redacted]0005US1
DATE: March 6, 2002

JC698 U.S. PTO
03/06/02

J1040 U.S. PTO
10/091774
03/06/02

Sir:

Transmitted herewith for filing under rule 1.53(f) is the Patent Application of:

Inventors: Sumio Morioka, Yasunao Katayama, Toshiyuki Yamane.

#4

ASSIGNEE NAME: International Business Machines Corporation
ASSIGNEE RESIDENCE: Armonk, New York

For: COMBINATIONAL CIRCUIT, AND ENCODER, DECODER AND SEMICONDUCTOR DEVICE USING THIS COMBINATIONAL CIRCUIT

Enclosed are:
__X__ 20 Sheets of Informal Drawings.
____ (Unsigned) Assignment of the invention to International Business Machines Corporation, Armonk, New York 10504.
__X__ Certified copies of two Japanese Patent application numbers 2001-196027 and 2001-066573.
__X__ (Unsigned) Declaration and Power of Attorney is attached to the application.
____ Associate Power of Attorney.
____ Information Disclosure Statement with form PTO-1449 with references attached.

The filing fee has been calculated as shown below:

(Col. 1)	(Col. 2)	SMALL ENTITY	OTHER THAN A	
FOR:	NO. FILED	NO. EXTRA	RATE	FEE
BASIC FEE				$ 740.00
TOTAL CLAIMS	22- 20 =	2	X $ 18 =	$ 36.00
INDEP CLAIMS	3 - 3 =	0	X $ 84 =	$ 0.00
_X_ MULTIPLE DEPENDENT CLAIM PRESENTED			+ $280 =	$ 280.00
			TOTAL	$ 804.00

If the difference in Col. 1 is less than zero, enter "0" in Col. 2.

__x__ Please charge my Deposit Account No. 09-0468 in the amount of $804.00.

__x__ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. 09-0468. A duplicate copy of this sheet is enclosed.

__x__ Any additional filing fees required under 37 CFR 1.16.
__x__ Any patent application processing fees under 35 CFR 1.17.

Respectfully submitted,

By Paul J. Otterstedt
Paul J. Otterstedt
Registration No.: 37,411
Tel. (914) 945-3158

IBM CORPORATION
INTELLECTUAL PROPERTY LAW DEPT.
P.O. BOX 218
YORKTOWN HEIGHTS, NY 10598

Express Mail Label Number:
EV091800887US
Date of Deposit: March 6, 2002

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　６月２８日

出願番号
Application Number: 特願２００１－１９６０２７

出願人
Applicant(s): インターナショナル・ビジネス・マシーンズ・コーポレーション

２００１年　７月２７日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号　出証特２００１－３０６６７７４

【書類名】　特許願

【整理番号】　JP9010005

【提出日】　平成13年 6月28日

【あて先】　特許庁長官　殿

【国際特許分類】　H04L 1/00

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　森岡　澄夫

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　片山　泰尚

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　山根　敏志

【特許出願人】

【識別番号】　390009531

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレーション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【代理人】

【識別番号】　100106699

【弁理士】

【氏名又は名称】　渡部　弘道

【復代理人】

【識別番号】　100110607

【弁理士】

【氏名又は名称】　間山　進也

【先の出願に基づく優先権主張】

【出願番号】　特願2001- 66573

【出願日】　平成13年 3月 9日

【手数料の表示】

【予納台帳番号】　062651

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9706050

【包括委任状番号】　9704733

【包括委任状番号】　0004480

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイス

【特許請求の範囲】

【請求項１】　ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）における符号化されたディジタル信号の２個以上の乗算を独立して行う複数の乗算器を含む組み合わせ回路において、

前記乗算器は、入力側ＸＯＲ演算部と、ＡＮＤ演算部と、出力側ＸＯＲ演算部とを含んで構成され、前記入力側ＸＯＲ演算部を、複数の乗算器が共有する、組み合わせ回路。

【請求項２】　複数の前記乗算器の入力が共通する、請求項１に記載の組み合わせ回路。

【請求項３】　前記組み合わせ回路は、波長多重通信により送信されるディジタル信号における誤り位置を算出する誤り位置算出部および誤り値算出部に使用される、請求項１または２に記載の組み合わせ回路。

【請求項４】　前記符号化されたディジタル信号から算出されるシンドロームが入力される請求項１～３のいずれか１項に記載の組み合わせ回路。

【請求項５】　復号または誤り訂正または暗号化のいずれかに使用される請求項１～４のいずれか１項に記載の組み合せ回路。

【請求項６】　暗号の符号化回路および復号化回路に使用される、請求項１または２に記載の組み合わせ回路。

【請求項７】　ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）における積和演算を行う組み合わせ回路において、それぞれの前記乗算器は、前記ＡＮＤ演算部と、前記出力側ＸＯＲ演算部との間に接続される加算器を含み、前記出力側ＸＯＲ演算部が共有され、複数の前記乗算器の前記ＡＮＤ演算部の出力を前記加算器により加算し、該加算結果を共有される前記出力側ＸＯＲ演算部により演算する、組み合わせ回路。

【請求項８】　複数の前記乗算器の入力が共通しており、前記入力側ＸＯＲ演算部が複数の前記乗算器により共有される、請求項７に記載の組み合わせ回路。

【請求項９】　前記組み合わせ回路は、波長多重通信により送信されるディジタル信号における誤り位置を算出する誤り位置算出部および誤り値算出部に使用される、請求項７または８に記載の組み合わせ回路。

【請求項１０】　前記符号化されたディジタル信号から算出されるシンドロームが入力される請求項７～９のいずれか１項に記載の組み合わせ回路。

【請求項１１】　復号または誤り訂正または暗号化を行うために使用される、請求項７～１０のいずれか１項に記載の組み合わせ回路。

【請求項１２】　暗号の符号化回路および復号化回路に使用される、請求項７～１０のいずれか１項に記載の組み合わせ回路。

【請求項１３】　請求項１～１２のいずれか１項に記載の組み合わせ回路を含む符号化装置。

【請求項１４】　請求項１～１２のいずれか１項に記載の組み合わせ回路を含む復号装置。

【請求項１５】　ディジタル信号を処理するために使用される半導体デバイスであって、該半導体デバイスは、

符号化された入力ディジタル信号を受信するための入力手段と、

前記符号化された入力ディジタル信号を処理して誤り位置多項式係数と誤り値多項式係数とを算出する処理手段と、

前記誤り位置多項式係数と、前記誤り値多項式係数とから誤りが訂正された出力ディジタル信号を出力する出力手段とを含み、

前記入力手段は、順序回路から構成され、前記処理手段は、組み合わせ回路から構成される、半導体デバイス。

【請求項１６】　前記組み合わせ回路は、ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）におけるディジタル信号の２個以上の乗算を独立して行う複数の乗算器を含み、

前記乗算器は、入力側ＸＯＲ演算部と、ＡＮＤ演算部と、出力側ＸＯＲ演算部とを含んで構成され、前記入力側ＸＯＲ演算部が、複数の前記乗算器に共有される、請求項１５に記載の半導体デバイス。

【請求項１７】　前記組み合わせ回路は、ガロア拡大体ＧＦ（2^m）（ｍは、

２以上の整数）における積和演算器を含み、かつそれぞれの前記乗算器は、前記ＡＮＤ演算部と、前記出力側ＸＯＲ演算部との間に接続される加算器を含み、前記出力側ＸＯＲ演算部が共有されていて、複数の前記乗算器の前記ＡＮＤ演算部の出力を前記加算器により加算し、該加算結果を共有される前記出力側ＸＯＲ演算部により演算する、請求１５に記載の半導体デバイス。

【請求項１８】　複数の前記乗算器の入力が共通しており、前記入力側ＸＯＲ演算部が複数の前記乗算器により共有される、請求項１５または１６に記載の半導体デバイス。

【請求項１９】　前記組み合わせ回路は、波長多重通信により送信されるディジタル信号の復号回路の誤り位置を算出する誤り位置算出部および誤り値算出部に使用される、請求項１５～１８のいずれか１項に記載の半導体デバイス。

【請求項２０】　前記半導体デバイスは、復号または誤り訂正または暗号化に使用される、請求項１５～１８のいずれか１項に記載の半導体デバイス。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスに関し、より詳細には、本発明は、高速の光通信分野において特に効果的に誤りを訂正することを可能とする、組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスに関する。

【０００２】

【従来の技術】

高速で高度な誤り訂正技術の重要性

インターネットの拡大とｅ－ビジネスの進展に伴い、コンピュータの扱うデータ量とスピードとは、加速度的に増加している。これに伴い、複数のコンピュータの間におけるデータ通信の速度を高めることが要求されており、近年では４０Ｇｂｐｓのデータ転送速度を使用する光通信の使用が普及しつつある。このような高速の、特に光通信といった通信方法において、システム・レベルの誤りの発

生率を現在と同等に保つためには、コンピュータの扱うデータ量に比例してデータ通信における信頼性をよりいっそう向上させる必要がある。

【０００３】

この信頼性を向上させる重要な技術としては、高度な数学を駆使してさまざまな要因（伝送経路でのノイズなど）による誤りを自動的に訂正する誤り訂正符号と呼ばれるものがある。その中でも、今日、多く使われている代表的な誤り訂正符号として、ハミング符号とリード・ソロモン符号を挙げることができる。上述したハミング符号は、基本的にはビット単位の誤り訂正を行なうものであり、その訂正能力は低いといえる。例えば、ハミング符号を使用すれば、1ビットの誤りを発見した場合は訂正するが、2ビットの誤りは検出するのみとなる。ハミング符号を用いる誤り訂正システムは、誤り訂正処理自体が簡単なため、誤り訂正処理を並列的に処理させることにより、１Ｇｂｐｓ（1秒間に10億ビット）を大きく超える高速処理が可能であることが知られている。

【０００４】

一方で、リード・ソロモン符号は、複数個の連続したビット単位（シンボル）の誤りを訂正することが可能で高度な訂正能力を有する優れた誤り訂正方法である。しかしながら、リード・ソロモン符号は、処理に複雑な演算を多用するため並列処理が難しく、例えば、8ビットのデータを100MHzでパイプライン処理を実行させたとしても、800Mb/sの処理速度しか得られないのが現状である。したがって、従来のリード・ソロモン符号の復号方式は、低速の通信分野、またはハードディスクやCD-ROM等の二次記憶装置の分野の製品など、現状では主にデータ処理速度の比較的低い分野で応用されており、今後高速性が要求される分野への応用には限界があった。

【０００５】

高速光通信分野で要求される誤り訂正技術

特にコンピュータおよびコンピュータが関連するデータ通信に使用される高速光通信の分野では、近年、ますます増大しつつあるインターネットをバックボーンとして、波長多重通信WDM(Wavelength Division Multiplexing)や、さらに波長多重度を向上させたDWDM(Dense WDM)を使った、テラビット／秒の高速通信シ

ステムが、所定の長さのフレームを連続して同期転送するSONETといった技術をベースに導入されつつある。

【０００６】

光通信を使用する上述したデータ通信における波長多重度の向上に伴い、近接する波長間でのクロストークが問題となる。このようなクロストークに対処するためこれまで、光を用いる波長多重通信における長距離(Long Haul)での転送に際しては、誤り訂正法としてFEC(Forward Error Correction)が使用されている。ITU(International Telecommunication Union) では、ITU-T G.975において、インターリーブされたm=8 (8ビット／シンボル)の(255、239)RS符号（符号長ｎ＝２５５バイト）の使用が標準化され、G.709では、FECのFrame構造を定めるDigital Wrapperの標準が採用されている。

【０００７】

このような例えばDigital Wrapper標準においては、これまでの低速なシリアル・リード・ソロモン復号回路を複数並列に並べることにより、必要な処理能力が確保されており、リード・ソロモン符号のインターリーブは必要不可欠な技術になっている。

【０００８】

高速かつ高度な誤り訂正技術における先行技術

これまで、上述したような光通信での必要性とは独立して、組み合わせ回路によるリード・ソロモン符号のパラレル高速復号の研究が行われてきている。

【０００９】

図１は、誤り訂正装置に使用することができるこのような高速復号回路の例を示す。

図１に示される復号回路は、これまでの１復号回路あたりの復号処理速度を10倍以上高速化すること、高度な誤り訂正能力を持つリード・ソロモン符号の誤り訂正処理を、ハミング符号と同程度の処理速度でパラレル復号を行なうことを実現するものである。図１に示した復号回路では、ガロア体上での対称関数による新たな表現形式を、リード・ソロモン符号の復号処理に応用することにより、誤り値を直接計算できるＯ（ｔ）次の誤り値多項式Ｅｒ(ｘ)が用いられている（ｔは

誤り訂正できる最大のシンボル数）。

【００１０】

図１に示した復号回路では、この多項式の使用により、シンドローム計算、誤り位置評価だけではなく、誤り値の評価も、これまでのようにForneyアルゴリズムを用いて２つの多項式の評価結果を除算して求めるのではなく、１つの多項式で直接計算可能とするので、はるかに簡単化することができる。さらに、図１に示した復号回路では、Ｅｒ(x)の係数計算だけではなく、誤り位置多項式Λ(x)の係数計算に対しても、組み合わせ回路に適した表現が採用され、必要な演算回路の数の削減を、高速化と同時に可能としている。

【００１１】

図１に示される復号回路を使用することにより、0.35μｍの標準的なASIC技術を使って半導体上に試作されたランダムな４バイト誤り訂正処理回路では、320ビット幅のデータを低レイテンシ(45 ns) で並列処理することが可能とされ、現在のシリアル復号回路の典型的な処理速度の800Mb/sの10倍近い7Gb/s（1秒間に70億ビット）以上の処理速度が達成されている。さらに、図１に示した復号回路に対して、大規模並列の誤り訂正処理回路に特化した新たな回路最適化アルゴリズムを使用し、回路共有化手法を用いることにより、回路規模が縮小できることが示されている。さらに、図１の復号回路は、外部制御回路やレジスタを使わない組み合わせ回路であるので、高速処理にもかかわらず消費電力も抑えることができるという利点がある。

【００１２】

しかしながら、図１に示した復号回路においても、光通信に必要とされる４０Gbpsの処理速度には及ばず、また通常の回路共有化手法を用いるだけでは回路規模が、ＩＴＵで標準化された８バイト誤り訂正に対応するためには１チップに実装できないほどに大規模になってしまうという問題があった。

【００１３】

図２は、従来用いられている低速の復号方式を使用する光通信用誤り訂正回路の概略構成を示した図である。光通信分野での通信速度の向上に伴い、従来の低速のシリアル・リード・ソロモン復号回路を並列に並べる方法は、ますます困難

になってきている。既存のRS復号回路では、せいぜい１Ｇｂｓの処理性能しかない。このため、図２に示した復号方式においては、低速のシリアル・リード・ソロモン復号回路を並列に並べることで必要な処理性能が実現されている。しかしながら、図２に示した従来方式は多くのリード・ソロモン復号回路を並べる必要が生じ、光通信のデータ転送速度に比例して、回路規模が大きくなってしまうという不都合がある。図３には、図２に示した復号方法を使用する場合の、回路規模と、データ転送速度とをプロットした図を示す。

【００１４】

図４には、さらに別の復号回路の従来例を示す（A. Patel、 IBM J. Res. Develp.、 vol. 30 pp. 259-269、 1986)。従来の復号方式では、シンドロームの計算・誤り位置の計算については多項式の評価ですむので、容易に高速化することができる。しかしながら、図４に示すように誤り値の計算でForneyアルゴリズムを用いるため、２つの多項式、すなわち誤り位置多項式の微分ｄΛ(ｘ)／ｄｘと、シンドロームと誤り位置多項式とから計算される誤り評価多項式Ω(ｘ)との評価の後に除算を実行することが必要となる。そして、これが出力の高速化を妨げるクリティカル・パスとなり、充分な高速化を行うことができないという不都合がある。

【００１５】

OC-768 SONETでは、復号回路の入出力インターフェースとして、ＩＴＵ－Ｇ７０９で決められた１６インターリーブを仮定すると、300MHz以上の高速動作が期待されているので、これは、重大な問題となる。図４に示される復号回路を使用し、クリティカル・パスに相当する除算をさらに細かくパイプライン化して出力を高速化しようとすることも試みられている。

【００１６】

しかしながら、いくら細かくパイプライン化したとしても図４に示した復号回路では、誤りのない位置でも除算を行なう必要があり、パイプラインの細分化に伴って回路規模・消費電力共に増大する。また、誤り位置にのみ除算を行なおうとすると、誤り位置を事前に計算する必要があり、誤り位置と誤り値の同時計算が行なえないという不都合もある。また、図４に示した復号回路においては、誤

りがあるか無いかによって、誤り値の出力に必要なサイクル数が違うので、SONETのような同期式フレームに載せて連続したデータを高速に入出力する必要のある場合には、誤りパターン（誤り数・位置）に依存せず一定のサイクルで高速に誤り値を出力することが困難である。

【００１７】

図５にはさらに別の従来の復号回路を示す。図１に示したパラレル・リード・ソロモン復号方式を光通信の分野に応用しようとすると、回路あたりの処理性能という観点では既存の方式よりも優れているためインターリーブされていないRS符号では問題なく応用可能である。しかしながら、ITU-T G.975で決められているインターリーブされたリード・ソロモン符号に対しては、高速で大きなバッファとセレクタを用いて信号の順序を並べ替える必要が生じるため、必ずしも効率のよいものではなかった。すなわち、(255、239)RS符号の符号長は、２０４０ビットであり、１６インターリーブした場合には、１６バイト入力255バイト出力のシリアル・パラレル変換回路と255バイト入力１６バイト出力のパラレル・シリアル変換回路を必要とし、高速化という点ではある程度の目的を達成することができるものの回路サイズが著しく大規模なものとなり、実用レベルで提供することが困難である。

【００１８】

また、上述した復号回路に使用される誤り位置および誤り値の算出においては、ガロア拡大体（ＧＦ）2^m上における演算を大量に、かつ高速で実行でき、さらには、実装可能な回路規模で処理を実行させることが要求される。従来、上述したガロア体における演算について従来行われてきた検討においては、いずれも単体の演算（乗算や除算）をいかに効率的に行うかに重点が置かれており、それらの演算を、特に組み合わせ回路を採用して数十～数百以上の演算を行うことについては、これまでほとんど検討がなされていないのが現状である。その理由は、種々推定することができるが、その１つとしては、復号演算は多くはこれまで順序機械によって実装され、組み合わせ回路を採用することは、得られる処理性能と回路規模の点で利点が少ないものと判断されていたこともある。

【００１９】

一方、誤り訂正のアルゴリズムについて考察すると、誤り位置の計算アルゴリズムにおいては、ガロア拡大体ＧＦ（2^m）上で定式化されたYule-Walker方程式がリード・ソロモン符号の計算において発生し、このYule-Walker方程式を効率よく処理することが高速化を達成し、かつ必要とされる回路サイズを可能な限り小さくするために必要とされる。Yule-Walker方程式を解くというアルゴリズムを実行させる場合、高速化の目的から組み合わせ回路で実現する場合には、謝り訂正能力の増大とともにYule-Walker方程式を解いて誤り位置を求める部分が組み合わせ回路の回路サイズの観点からみて非常に大きな割合を占めることになる。

【００２０】

加えて、リード・ソロモン符号の復号化を組み合わせ回路を用いて実現し、実際のシステムへと適用する場合には、処理に対する汎用性を付与し、付加的な回路または処理を追加しないためにも、任意の最小距離を持つリード・ソロモン符号の復号化に適用できるアルゴリズムを提供することが望ましい。特に光通信の分野ではITU(International Telecommunication Union)において(255、239) リード・ソロモン符号を使用することが標準化されたため、訂正可能な誤りの最大個数が8、最小距離が17である場合にも、リード・ソロモン符号の復号化を効率的に行うことが可能なアルゴリズムが必要とされている。

【００２１】

Yule-Walker方程式を解くという数学的な問題を、組み合わせ回路を用いてハードウエアとして実装可能な規模で実現するためには、回路規模の増大を抑制し、乗算器の個数を削減するアルゴリズム、およびこのアルゴリズムを効率よく処理する組み合わせ回路構成が必要とされている。すなわち、上述した誤り訂正装置および誤り訂正のアルゴリズムを、高速化といった目的を達成しつつ、許容可能な回路サイズとしてデバイス化を可能とする組み合わせ回路が必要とされていた。

【００２２】

【発明が解決しようとする課題】

本発明は上述した従来の技術における不都合に鑑みてなされたものであり、本

発明は、高速（４０Ｇｂｐｓもしくはそれ以上）の光通信分野、さらにより限定的には連続したデータを同期フレームとして転送するSONET上での波長多重通信のため、インターリーブされたリード・ソロモン符号のための効率的な組み合わせ回路、該組み合わせ回路を使用する信号処理装置、および半導体デバイスを提供することにある。

【００２３】

すなわち、本発明は、回路規模あたりの処理性能が高い（低レイテンシ・高スループット）、組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提供することを目的とする。

【００２４】

さらに、本発明の別の目的は、インターリーブされたリード・ソロモン符号にも上述した特長を失うことなく対応できる構成の、柔軟な組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提示することにある。

【００２５】

さらに本発明の別の目的は、インターリーブされた受信語のそれぞれの誤りパターン（誤り数、位置）に関係なく一定のサイクルで誤り語を連続して高速に出力する組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提供することにある。

【００２６】

さらに、本発明の別の目的は、ガロア拡大体ＧＦ（2^m）（ｍは２以上の任意の自然数）上の演算回路のうち、いくつかの入力が共通な複数の乗算（例えばＡＢとＡＣとＡＤ）を行う回路、および積和演算ＡＢ＋ＣＤ＋ＥＦ＋…を行う回路を、高速かつ高効率で処理でき、小規模な回路によって実現することが可能な組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提供することにある。

【００２７】

さらに、本発明の別の目的は、リード・ソロモン符号の復号化を組み合わせ回路を用いて実現し、実際のシステムへと適用する場合に、処理に対する汎用性を

付与し、付加的な回路または処理を追加せずに任意の最小距離を持つリード・ソロモン符号の復号化に適用できる組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提供することにある。

【００２８】

【課題を解決するための手段】

本発明の上記課題は、本発明の組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提供することにより解決することができる。

【００２９】

本発明によれば、ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）における符号化されたディジタル信号の２個以上の乗算を独立して行う複数の乗算器を含む組み合わせ回路において、

前記乗算器は、入力側ＸＯＲ演算部と、ＡＮＤ演算部と、出力側ＸＯＲ演算部とを含んで構成され、前記入力側ＸＯＲ演算部を、複数の乗算器が共有する、組み合わせ回路が提供できる。本発明の組合せ回路は、複数の前記乗算器の入力が共通する。前記組み合わせ回路は、波長多重通信により送信されるディジタル信号における誤り位置を算出する誤り位置算出部および誤り値算出部に使用される。前記符号化されたディジタル信号から算出されるシンドロームが入力される。本発明の組み合わせ回路は、復号または誤り訂正または暗号化のいずれかに使用される。また、本発明の組み合わせ回路は、暗号の符号化回路および復号化回路に使用される。

【００３０】

本発明によれば、ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）における積和演算を行う組み合わせ回路において、それぞれの前記乗算器は、前記ＡＮＤ演算部と、前記出力側ＸＯＲ演算部との間に接続される加算器を含み、前記出力側ＸＯＲ演算部が共有され、複数の前記乗算器の前記ＡＮＤ演算部の出力を前記加算器により加算し、該加算結果を共有される前記出力側ＸＯＲ演算部により演算する、組み合わせ回路が提供される。本発明の組み合わせ回路は、複数の前記乗算器の入力が共通しており、前記入力側ＸＯＲ演算部が複数の前記乗算器によ

り共有される。前記組み合わせ回路は、波長多重通信により送信されるディジタル信号における誤り位置を算出する誤り位置算出部および誤り値算出部に使用される。本発明の組み合わせ回路には、前記符号化されたディジタル信号から算出されるシンドロームが入力される。本発明の組み合わせ回路は、復号または誤り訂正または暗号化を行うために使用される。

【００３１】

また、本発明によれば、上述した組み合わせ回路を含む符号化装置または復号装置。が提供される。

【００３２】

さらに本発明によれば、ディジタル信号を処理するために使用される半導体デバイスであって、該半導体デバイスは、

符号化された入力ディジタル信号を受信するための入力手段と、

前記符号化された入力ディジタル信号を処理して誤り位置多項式係数と誤り値多項式係数とを算出する処理手段と、

前記誤り位置多項式係数と、前記誤り値多項式係数とから誤りが訂正された出力ディジタル信号を出力する出力手段とを含み、

前記入力手段は、順序回路から構成され、前記処理手段は、組み合わせ回路から構成される、半導体デバイスが提供される。本発明においては、前記組み合わせ回路は、ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）におけるディジタル信号の２個以上の乗算を独立して行う複数の乗算器を含み、

前記乗算器は、入力側ＸＯＲ演算部と、ＡＮＤ演算部と、出力側ＸＯＲ演算部とを含んで構成され、前記入力側ＸＯＲ演算部が、複数の前記乗算器に共有される。本発明においては、前記組み合わせ回路は、ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）における積和演算器を含み、かつそれぞれの前記乗算器は、前記ＡＮＤ演算部と、前記出力側ＸＯＲ演算部との間に接続される加算器を含み、前記出力側ＸＯＲ演算部が共有されていて、複数の前記乗算器の前記ＡＮＤ演算部の出力を前記加算器により加算し、該加算結果を共有される前記出力側ＸＯＲ演算部により演算することができる。本発明の半導体デバイスは、複数の前記乗算器の入力が共通しており、前記入力側ＸＯＲ演算部が複数の前記乗算器によ

り共有される。前記組み合わせ回路は、波長多重通信により送信されるディジタル信号の復号回路の誤り位置を算出する誤り位置算出部および誤り値算出部に使用される。本発明の前記半導体デバイスは、復号または誤り訂正または暗号化に使用される。

【００３３】

【発明の実施の形態】

以下本発明を、図面に示した実施の形態に基づいて説明するが、本発明は、図面に示した実施の形態に限定されるものではない。

【００３４】

セクション１　＜復号回路＞

図６には、光通信により送信されたディジタル信号に含まれる誤りを訂正するために使用することができる本発明による復号回路を示す。図６に示した復号回路は、入力部１０と、処理部１２と、出力部１４とを含んで構成されている。入力部１０は、例えば１６インターリーブされた入力ディジタル信号ＩＤが入力される。処理部１２は、この入力部からの出力を受け取って処理を行ない、誤り位置多項式係数と誤り値多項式係数とを算出する。出力部１４は、この処理部１２からの出力を受け取り、出力された誤り位置と誤り値とをＡＮＤ処理し、入力される入力ディジタル信号ＩＤとＸＯＲ処理を行なって、出力ディジタル信号ＯＤを生成する。出力ディジタル信号ＯＤは、入力ディジタル信号に含まれる可能性のある誤りが訂正されている。

【００３５】

図６に示した復号回路に入力される入力ディジタル信号ＩＤとしては、本発明においては、光通信、特に４０Ｇｂｐｓといった高速のデータ転送レートで波長多重通信法により送信されるディジタル信号を入力することができる。より具体的には、上述した入力ディジタル信号ＩＤは、例えば（２５５，２３９）ＲＳ符号により符号長２０４０ビットとして送信されるものを用いることができる。通常、波長多重通信方においては、上述した入力ディジタル信号は、インターリーブ方式が採用されて、例えば１６の並列な２５５バイト・ストリームとして本発明の復号回路に入力される。

【００３６】

図６に示した本発明の復号回路においては、上述した入力ディジタル信号ＩＤは、入力部１０においてインターリーブされて並列に入力され、それぞれの入力について、受信多項式が規定され、この受信多項式からシンドロームＳｉが算出され、入力部１０の出力とされる。入力部１０（シンドローム算出部）の出力であるシンドロームＳｉは、（２５５，２３９）ＲＳ符号の場合は、入力ディジタル信号２５５バイトから得られる１６バイトのディジタル情報として生成される。例えば、本発明の図６において説明する復号回路の実施の形態においては、入力ディジタル信号は、２０４０ビットが１６インターリーブされて１６の２５５バイト・シリアル・ストリームとして入力部１０へと入力され、入力部１０において１６のシリアル・ストリームに対応する１６の１６バイト・シンドロームが生成されることになる。

【００３７】

図６に示されるように、入力部１０においては、それぞれの入力ディジタル信号ＩＤのシリアル・ストリームＩＤＳに対して１つのシンドローム算出部１６が割り当てられ、シンドロームが算出される構成とされている。算出されたそれぞれのシリアル・ストリームＩＤＳについてのシンドロームは、レジスタ１８に保持された後、出力部１４へとマルチプレクサにより符号語間でシリアル化されたデータとして出力される。例えば、上述したように、１６インターリーブされた２５５バイトの入力ディジタル信号から得られた１６バイトのシンドロームからは、誤り位置および誤り値の算出の為に、１２８ビットの信号が得られることになる。図６に示した入力部１０において使用することができるシンドローム算出部１６としては、順路回路を使用する回路など、これまで知られたいかなる回路でも使用することができる。なお、シンドロームの定義および算出方法についてはより詳細に後述する。

【００３８】

図６に示された復号回路においては、算出されたそれぞれのシンドロームS_iは、処理部１２へとそれぞれ順次入力される。図６においては説明の便宜のため、処理部１２は、１つだけを例示して示している。図６に示した処理部１２は、

複数の乗算器により構成された組み合わせ回路から構成される、誤り位置多項式Λ（ｘ）を算出するための誤り位置多項式係数算出部１８と、誤り値多項式Ｅｒ（ｘ）を算出するための誤り値多項式係数算出部２０と、を含んで構成されている。処理部１２において使用される乗算器を含む組み合わせ回路の詳細についてはセクション２＜組み合わせ回路＞においてより詳細に説明する。

【００３９】

図６に示した処理部１２の出力である誤り位置多項式Λ（ｘ）と、誤り値多項式Ｅｒ（ｘ）とは、出力部１４へと、図示しないデマルチプレクサにより、例えばインターリーブの数に対応する用にデマルチプレクスされた後、入力される。出力部１４には、それぞれの入力ディジタル信号のインターリーブの数に対応する数だけ配置されたレジスタ２２と、ＡＮＤゲート２４と、ＸＯＲゲート２６とが形成されている。出力部１６では、それぞれのシンドロームから得られた誤り位置情報（誤り位置には「１」、そうでなければ「０」）Λ_{eval}を使って、誤り値E_rをＡＮＤゲート２４により選択する。さらに選択された出力は、ＸＯＲゲート２６において加算が行われる。このＸＯＲ選択ゲート２６には、符号化された入力ディジタル信号ＩＤから得られるシリアル・ストリームＩＤＳが、ｉｍｎビットのバッファ２８ａ，２８ｂを介してＸＯＲゲート２６へと入力されており、ＸＯＲゲート２６においてガロア拡大体ＧＦ（2^m）上の減算を実行させることによって誤りが除去された２５５バイトの出力ディジタル信号ＯＤとして与える構成とされている。

【００４０】

図６に示した本発明の復号回路においては、上述した処理部１２を構成する組み合わせ回路を順序回路として構成することもできるが、本発明において特に複数の乗算器を、入力側ＸＯＲ演算群（変数前処理部）と、ＡＮＤ演算群、出力側ＸＯＲ演算群（剰余演算）の３段構成とし、変数前処理部または剰余演算部の一方ないし両方を、複数の乗算器間で共有させる構成を採用することにより、従来誤り位置および誤り値の算出においてクリティカル・ポイントとされてきた処理部１２における回路規模を実用上許容可能な規模としつつ、効率的に乗算器から構成させることが可能となる。

【００４１】

図６に示した本発明の復号回路では、上述したように出力部１４を、処理速度を低下させる非線形演算を実行させる回路構成を採用することなく、定数乗算器と加算器いった線形演算を行う回路のみで構成させる。このため、本発明の復号回路は、処理速度を低下させず高速に、かつ回路構成を小規模としつつ復号回路として構成することが可能となる。さらに本発明者らは、鋭意検討の結果、特定の構成を含む乗算器から構成される組み合わせ回路と、該組み合わせ回路において誤り位置および誤り数を効率的に算出することを可能とするアルゴリズムとを採用することにより、従来にまして柔軟で高速かつ、小規模の復号回路を構成することを見出し、本発明に至ったものである。

【００４２】

本発明において採用する誤り位置多項式係数算出の方法またはアルゴリズムについてはより詳細に組み合わせ回路構成と共に後述するが、以下に、本発明の復号回路における処理部１２に含まれる誤り値多項式係数算出部２０の機能・作用について説明する。

【００４３】

誤り値計算アルゴリズムの選択

本発明の復号アルゴリズムについては、誤り位置の評価だけではなく誤り値の評価もＯ（ｔ）次の多項式計算（線形演算）で直接計算可能なアルゴリズムをインターリーブされたリード・ソロモン符号の復号に対して適用するものである。この際、本発明において採用するアルゴリズムにおいては、特に誤り値の計算に必要な除算を、多項式の評価の後の出力のクリティカル・パス中で誤り位置ごとに行うのではなく、多項式の評価の前に符号語ごとにただ一度線形演算で行う。これにより、多項式評価の値を誤りの値として一定のサイクルで直接高速出力することが可能となる。

【００４４】

さらに、誤り値Ｅｒ(x)多項式の次数は、ｔ個の独立した出力を出すのに最低必要とされるｔ－１次まで低くできることが判明している。その際に得られるｔ個の係数は、誤り位置多項式の係数とシンドロームとを使って計算することが可

能である。このアルゴリズムの採用により、シンドローム計算、ならびに、誤り位置評価だけではなく、誤り値の評価も線形演算回路だけで実行することが可能となり、入出力出力回路全体を簡略化高速化することが可能となる。

【００４５】

本発明の復号アルゴリズムまたは復号方法において種々の誤り値多項式を使用することができるが、以下、ｅ個の誤りがｉ$_0$，．．．，ｉ$_{e-1}$に発生したときの誤り値多項式Ｅｒ(ｘ)が下記式

$$Er^{(e)}(x)=\sum_{l=0}^{e-1}\frac{E_{i_l}\prod_{j\neq l}(x+a^{i_j})\prod_{j,k\neq l,j<k}(a^{i_j}+a^{i_k})}{\prod_{j<k}(a^{i_j}+a^{i_k})}$$

の形式で与えられ、除算を含むが、分母は多項式ではなく、符号語ごとに定数である実施の形態を例として、本発明の機能・作用について説明する（上記式中、ａは、ガロア拡大体の原子元を意味する）。

【００４６】

上述した誤り値多項式Ｅｒ$^{(e)}$(ｘ)は、多項式Ｅｒ$^{(e)}$(ｘ)の計算に位置

$$a^{i_k}$$

（以下ａ^{ik}と略する）に存在するｋ番目の誤り値

$$E_{i_k}$$

（以下Ｅ$_{ik}$と略記する）が必要となるので本末転倒であり、直接本発明の復号回路に使用することができない。しかしながら、もしＥｒ$^{(e)}$(ｘ)中のすべてのＥ$_{ik}$といくつかのａ^{ik}を、シンドロームＳ$_i$を使って記述すれば、回路化が可能となる。さらに、この場合Ｅｒ$^{(e)}$(ｘ)中のすべてのａ^{ik}を誤り多項式Λ$^{(e)}_j$を使用して記述することで、誤り位置を求める前に誤り値の算出を実行することが可能となる。したがって誤り値の算出を並列化することが可能となり、この結果高速化を達成することができる。

【００４７】

上述したプロセスは、以下の復号アルゴリズムまたは復号方法を使用すること

により実行することができる。まず、分母の部分を$\Lambda^{(e)}_j$で記述する。誤り値と誤り位置多項式の係数は、定数ファクタを除いて誤り位置、

$$a^{i_0}, a^{i_1}, a^{i_2}, \ldots, a^{i_{t-1}}$$

について、それぞれElementary symmetric functionとなる。例えば、誤り位置多項式の係数

$$\Lambda_1, \Lambda_2, \ldots, \Lambda_t$$

については、

$$\Lambda_1 = \sum_k a^{i_k}$$
$$\Lambda_2 = \sum_{k<l} a^{i_k} a^{i_l}$$

$$\vdots$$

$$\Lambda_{t-1} = a^{i_0} a^{i_1} a^{i_2} \cdots a^{i_{t-1}}$$

となっており、誤り位置、

$$a^{i_0}, a^{i_1}, a^{i_2}, \ldots, a^{i_{t-1}}$$

は、互いに交換可能である。また、Shur関数と呼ばれる２つのVandermonde行列式の割り算で定義されるものと、上述したElementary symmetric functionとの間に成り立つ関係式(例えば、I.G.Macdonald, Symmetric Functions and Orthogonal Polynomials, American Mathematical Society, 1998参照)と、ガロア拡大体ＧＦ（2^m）において成り立つ加算と２乗算の下記交換関係、

$$(a+b)^2 = (a-b)^2 = a^2 + b^2$$

から、以下の新規な関係が導出できる。

$$f^{(e)} = \prod_{m>l}(a^{i_m} + a^{i_l})$$

$$= \begin{vmatrix} \Lambda_1^{(e)}\Lambda_3^{(e)} & \cdots 0 & \cdots & 0 \\ 1 & \Lambda_2^{(e)}\Lambda_4^{(e)} & \cdots & 0 \cdots 0 \\ 0 & \Lambda_1^{(e)}\Lambda_3^{(e)} & \cdots & 0 \cdots 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots 0 & \cdots \Lambda_{e-2}^{(e)} & \Lambda_e^{(e)} \\ 0 & \cdots 0 & \cdots \Lambda_{e-3}^{(e)} & \Lambda_{e-1}^{(e)} \end{vmatrix}.$$

【００４８】

上記行列式を採用することにより、さきのＥｒ（ｅ）（ｘ）の分母を，Λ（ｅ）ｉで記述することが可能となる。Ｅｒ（ｅ）（ｘ）の分子についても同様に算出してみると、下記式で示すように、Ｅｒ（ｅ）（ｘ）の係数がＳｉ、Λ（ｅ）ｉのみで記述でき、E_{ik}と、a^{ik}を使用しなくとも算出することができることになる。

$$Er^{(e)}(x)$$

$$= \frac{\sum_{k=0}^{e-1} E_{i_k} \prod_{j\neq k}(x + a^{i_j}) \prod_{m,l\neq k,m>l}(a^{i_m} + a^{i_l})}{\prod_{m>l}(a^{i_m} + a^{i_l})}$$

$$= \frac{\sum_{k=0}^{e-1} E_{i_k} (\sum_{j=0}^{e-1} x^j \Lambda_{e-j-1\, i_k}^{(e)}) f^{(e-1)}(\Lambda_{1i_k}^{(e)}, \Lambda_{2i_k}^{(e)}, \ldots, \Lambda_{e-1\, i_k}^{(e)})}{f^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \ldots, \Lambda_e^{(e)})}$$

$$= \frac{\sum_{k,j,m=0}^{e-1} E_{i_k} a^{i_k m} x^j Er_{jm}^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \ldots, \Lambda_e^{(e)})}{f^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \ldots, \Lambda_e^{(e)})}$$

$$= \frac{\sum_{j,m=0}^{e-1} S_m x^j Er_{jm}^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \ldots, \Lambda_e^{(e)})}{f^{(e)}(\Lambda_1^{(e)}, \Lambda_2^{(e)}, \ldots, \Lambda_e^{(e)})}.$$

ここで、

$$\Lambda_{j\, i_k}^{(e)} = \Lambda_j^{(e)} + a^{i_k} \Lambda_{j-1\, i_k}^{(e)}$$

であり、具体的には、a^{ik}を除いた誤り位置に対応する誤り位置多項式の係数である。

【００４９】

上述したように、上述した関係を新たに採用することにより、非線形演算回路を使用することなく、ガロア拡大体ＧＦ（2^m）上における線形演算回路により

、処理部１４を構成することが可能となり、高速化を実現することが可能となる。

【００５０】

以下、上述の誤り値計算アルゴリズムを用いてインターリーブ符号にも対応した高速、小回路規模の符号回路を構成する為に以下の構成を採用する。

（１）高速の入力部・出力部（線形演算回路）の採用

まず、符号の構成（インターリーブ数）と入出力インターフェースのバス幅、クロック数等を考慮して、入力に接続されたシンドローム計算のための多項式評価（線形演算）回路、出力側に接続された誤り位置評価・誤り値評価のための多項式評価（線形演算）回路を、RS符号が巡回符号であることに起因するサイクリックな構造を生かして、1からnの任意のクロック数で(n,k)リードソロモン符号の前処理・後処理する高速な順序回路として実現したことである。特に、従来方式でクリティカル・パスであった誤り値評価の部分に対しても上述した構成を採用するこちが本発明においては有効である。また、本発明の構成によれば、符号語あたりの入力されるディジタル信号幅が２５５バイトの場合ばかりではなく、１、３、５、１５、１７、５１、８５バイトインターフェースとして、柔軟に対応することができる復号回路を提供することができる。

【００５１】

表１には、本発明の復号回路において入力ディジタル信号の入力幅に対して要求される処理クロックの関係を示す。

【００５２】

【表１】

表１：符号語あたりの入出力幅とクロック数（n=255）

入出力幅（バイト）	1	3	5	15	17	51	85	255
符号語（n＝255）処理に必要なクロック数	255	85	51	17	15	5	3	1

表１に示されるように、ディジタル信号の入力幅が小さくなればそれに対応して必要なクロック数は増加するものの、本発明の復号回路を採用することにより柔

軟に対応することが可能となることが示されている。

【００５３】

さらに、本発明においては、符号語あたりの入出力のバイト幅は表１で示された以外でも任意に選択可能である。例えば入出力幅８バイトの場合でも、符号の長さを最後にダミー１バイトを加えてn=256バイトとすることにより対応可能となる。

【００５４】

（２）非線型演算回路１２との接続

本発明においてはさらに、入力部１０および出力部１４として順序回路を用いて構成し符号のインターリーブ数に応じて複数用意し、前後の順序回路の間にシンドロームならびに誤り位置・誤り値多項式の係数を保持する回路と、マルチプレクサ・デマルチプレクサとを用いて、誤り位置・誤り値多項式の係数計算を行う非線型演算回路１２に接続する。この演算回路は上述したように、非線形演算を実行する乗算器といった非線形演算回路の組み合わせ回路として構成されている。このため、本発明においては、例えば、OC-768を仮定して(255,239)リードソロモン符号を１６インターリーブして使用する場合には、前後に１６個づつ順序回路を用意することが必要となる。すなわちこの場合、１６:1のマルチプレクス・デマルチプレクスを行う必要がある。しかしながら、本発明においては、シンドローム多項式の係数をマルチプレクス・デマルチプレクスするために、2040ビットではなく、１２８ビット（Ｅｒ（ｘ）の取り方によっては、後段は、136ビット）の信号を扱うだけですみ、マルチプレクサ・デマルチプレクサ、バッファの数共に大きく削減することが可能となる。

【００５５】

（３）中央の非線型演算回路を時分割に使う３段パイプライン動作方式

本発明においては、さらに、復号回路全体を線形演算回路（シンドローム計算）－非線型演算回路（誤り位置・誤り値多項式の係数計算）－線形演算回路（誤り位置・誤り値の評価）の３段パイプラインとして動作させ、低レイテンシの非線型演算回路をインターリーブされたそれぞれの符号の計算にシンドロームを符号語間でシリアルに順次与える事により時分割して用いる。このため、回路規模

あたりの処理能力の高い高効率のインターリーブ符号の復号動作を実現できる。例えば、OC-768のケースでは、中央の非線型演算回路は、上述した３段パイプラインの動作をさせるためには、約40nsのレイテンシでインターリーブされたそれぞれの符号語あたりの処理を終える必要があるが、最先端の半導体技術（0.18μm以上）を使い、組み合わせ回路を用いた実装を行うことにより、上述した復号回路をＡＳＩＣといった半導体バイスとして実現できる。

【００５６】

すなわち、本発明の復号回路は、（１）高速の入力部１０と高速の出力部１４を採用すること、特に従来では非線形演算を行う必要があった出力部１４に対して、誤り位置および誤り値を線形演算回路だけを使用して算出できるようにし、さらに、処理部１２に対しては上述した復号アルゴリズムに適合すると共に、回路サイズを低減させる乗算器構成を採用すること、（２）本質的に非線形演算を実行する処理部１２を、時分割して使用して、３段パイプライン動作方式を採用すること、による相乗的な効果により、高速かつ回路サイズが許容範囲の復号回路、誤り訂正装置を提供することを可能とするものである。以下、セクション２として、本発明の復号回路における処理部１２に含まれる乗算器から構成される組み合わせ回路について詳細に説明する。

【００５７】

セクション２＜組み合わせ回路＞

本発明の復号回路に使用する処理部１２は、線形演算回路、具体的には乗算器を使用した組み合わせ回路として構成される。本発明において使用される乗算器は、しかしながら、従来の組み合わせ回路に使用される乗算器は、ガロア拡大体ＧＦ（2^m）における乗算をＡＮＤ演算群の次にＸＯＲ演算群を行う２段階の構成とするのではなく、ＸＯＲゲート－ＡＮＤゲート－ＸＯＲゲートの３段階の構成を採用する。

【００５８】

単体の並列乗算回路の構成方式

単体の乗算回路に関しては従来より多くの検討がなされているものの、順序回路でなく組み合わせ回路として構成された並列乗算回路（Mastrovito Multiplier

）については意外に研究の歴史が浅く、近年検討が開始されたといってもよい。従来の並列乗算回路（以下、本明細書においては、単に乗算回路という。）の構成方式は、AND-XOR形式とXOR-AND-XOR形式の２種類で、相互変換可能である。ただし単体の乗算しか回路化しない場合には、AND-XOR形式のものが通常用いられる。その理由は、AND-XOR形式はよく研究されていて小規模回路を得る方法がかなり検討されているのに対し、XOR-AND-XOR形式については一般に回路規模が削減される保証がなく（むしろ増える場合がある）、設計作業の複雑化に見合うだけの削減効果がないと考えられることが、その理由であると考えられる。以下、各場合について検討を加える。

【００５９】

（１） AND-XOR形式

この方法は、筆算どおりに計算を進める教科書的な方法で、通常はこの形の回路を使用する。具体的には、乗算の引数となる２つの（ｍ－１）次多項式に対し、その係数どうしを組み合わせてm^2個の部分積をまず作る。これがAND部の処理内容である。次に、それらの部分積のうち次数が同じものどうしを加算して（２ｍ－２）次多項式を構成し、既約多項式による剰余演算を行って（ｍ－１）次の解を得る。これらがXOR部の処理内容である。ANDの個数はm^2、XORの個数はＯ（m^3）であるが、既約多項式や基底を選択することによりXORが（m^2-1）個で構成できることが広く知られている。任意の乗算回路は必ずこの形式で構成できる。

【００６０】

（２） XOR-AND-XOR形式

上述したAND-XOR形式の回路に対し、ブール代数の規則である（A and B）xor (A and C) = A and (B xor C)のもとで、剰余演算部にあるXOR演算をANDの前に移動して変数前処理演算部（入力側ＸＯＲ演算）とすることが、一般に可能である。これによって乗算器の回路を、XOR-AND-XOR形式とすることができる。XORの移動にあたって、A xor A = 0、B xor 0 = Bの性質を利用し、剰余演算部（出力側ＸＯＲ演算）のXOR中に同一の冗長タームを偶数個追加しておくことで、より多くのXOR演算を変数前処理部へ移動できる場合がある。この操作が可能なため

、XORの移動には、単に分配律をそのまま適用するのみならず多様なやり方が存在し得る。したがって、たとえ同一の基底や既約多項式のもとであっても、XOR-AND-XOR形式は複数存在することになる。ゲート数は、XOR-AND-XOR形式にすることでAND-XOR形式より増加する場合も減少する場合もあり、まちまちである。また、次に述べるComposite Field Multiplierのように、特殊な基底のもとでシステマティックにXORゲートを削減する方法もこれまでに知られている。

【００６１】

（３）限定された体のみに適用可能なXOR-AND-XOR形式の構成法（Composite Field Multiplier）

Composite Field Multiplierは、ｍが合成数で、なおかつ体の要素の表現に用いる基底が通常の基底（多項式基底や正規基底）でなくてもよいという、特殊な場合に限って使える乗算回路構成法である。以下、より詳細に説明を行う。ｍが合成数のとき、ＧＦ（２）から２回以上の体の拡大によって拡大体ＧＦ（2^m）を構成できる。その拡大の過程に従って再帰的な構造の乗算回路を構成するのがComposite Field Multiplierである。このとき、例えば１回の２次拡大に対し、ＧＦ（2^m）上の２つの値Ａｘ＋ＢとＣｘ＋Ｄ（ここでＡ，Ｂ，Ｃ，Ｄは、それぞれ部分体ＧＦ（$2^{m/2}$）上の値）の積が、

$$(Ax+B)(Cx+D)=ACx^2+((A+B)(C+D)+AC+BD)x+BD)$$

であることを用いれば、部分体上の乗算を４個から３個へ減らすことができ、回路規模を減少させることができる（KOA）。同時に、回路はXOR-AND-XORの構造として構成することができる（乗算の前に行う加算が、ANDの前に配置されるＸＯＲに対応する）。なお、KOAの使用はComposite Field Multiplierであることが前提であって、そうでない乗算回路ではKOAは通常使用することができない。また、仮にｍが同手法を適用できる値であっても、体の変換回路が必要でそのオーバーヘッドのためにかえって回路規模が増えるので、単体の乗算しか回路化しない場合には同手法は通常採用される構成とはならないものである。

【００６２】

＜本発明における組み合わせ回路の通常の乗算器による構成＞

本発明が対象とする入力共通乗算回路群、および積和演算回路を通常のAND-XOR構成で構成した場合の構成を図７および図８に示す。図７においては組み合わせ回路の例として、２つの乗算器を使用する組み合わせ回路が示されている。図７に示されるように従来の乗算器は、第１の入力Ａ１が乗算器４０および乗算器４２へと入力され、乗算器４０には、第２の入力Ｂ１が入力されて第１の出力４５が、乗算器４２には第３の入力Ｂ２が入力されて第２の出力４６とが出力されている。図７に示すとおり、従来の構成では、乗算間で共通な入力があっても、そのために共有可能となる回路はまったく存在しない。図８には、積和演算を行うための組み合わせ回路を従来の乗算器の構成により構成した従来例を示す。図８に示す積和演算を行うための組み合わせ回路においても、そのままでは共有可能な回路は存在しないことがわかる。

【００６３】

図９は、従来の構成の乗算器を使用する組み合わせ回路の例を示す。図９の入出力では、１シンボルを１本の線として表記してある。また、入出力はそれぞれ８ビット幅であるものと仮定している。図９における組み合わせ回路においては、６シンボルの入力と１シンボルの出力とを含み、さらに７個の乗算回路と５個の加算回路を含んで構成されている。図９に示した組み合わせ回路においては、Ｓ０，．．．，Ｓ３Ｑで示された入力が乗算器群４６へと入力され、加算器群４８により加算された後、積和演算回路５０へと入力されて、各入力の積和演算の結果である出力Ｌ２１Ｑが生成されている。

【００６４】

図９においては、破線で示したクロスターム構成演算と剰余演算の組み合わせが１つの乗算に相当する。図９において示されている乗算器の回路は標準的なものなので、詳細な回路の説明については省略する。この乗算回路を含む組み合わせ回路は、64AND+約103XORの数のゲートを含んでおり、回路全体のゲート数は448AND+約761XORとなる。図９から明らかなように、ほとんどの乗算の入力は、その一方あるいは両方が他の乗算と共通である。また、最終段などで積和演算が行われている。

【００６５】

表２には、ガロア拡大体ＧＦ（2^8）において従来の標準的なＡＮＤ－ＸＯＲの２段構成の乗算回路、Composite Field Multiplierおよび部分体拡大体ＧＦ（2^4）の乗算をＸＯＲ－ＡＮＤ－ＸＯＲにする改変を行った乗算回路に含まれる各ゲートの数を示す。

【００６６】

【表２】

	変数前処理		クロスターム構成		剰余演算
標準的なAND-XORの乗算回路	なし	+	64AND	+	103XOR
Composite Field Multiplier	4 XOR*2個	+	48AND	+	56XOR
本発明の効果を挙げるために、部分体拡大体GF(2⁴)の乗算をXOR-AND-XORにしたもの					
a. 3個中3個の部分体乗算を改変	22 XOR*2個	+	30AND	+	44XOR
b. 3個中2個を改変	16 XOR*2個	+	36AND	+	48XOR
c. 3個中1個を改変	10XOR*2個	+	42AND	+	52XOR

【００６７】

すなわち、単体の乗算器のみに着目すれば、上記a～cいずれも回路規模は従来のComposite Field Multiplierの場合より増加しており，最小規模のものではない。このように、単に乗算器をＸＯＲ－ＡＮＤ－ＸＯＲの３段構成とするのでは、回路の規模を逆に大きくする場合もある。

【００６８】

＜本発明の組み合わせ回路における乗算器構成＞

一般には、多数の乗算や積和演算が絡み合うとブール代数上での最適化は困難である。しかしながら、本発明における処理部１２として組み合わせ回路を使用することを考慮すれば、上述した積和演算や乗算が並列かつ多段に多数接続された構造となっており、ｉ段目の積和演算は一般に０段目～ｉ－１段目の出力を入力とする。したがって、後段の演算回路になるとほぼ組み合わせ回路の回路全体において、共通した入力を並列処理する必要が生じるため、最適化範囲が広がることになる。本発明者らは、この点に着目し、他の演算とのバランスも取りつつ、ブール代数上の最適化することにより、乗算器の効率的な組織化を達成したものである。

【００６９】

図１０は、図７に示した従来の構成の乗算器および加算器からなる組み合わせ回路を、乗算器を３段構成として本発明の組み合わせ回路とした本発明の実施の形態の組み合わせ回路を示す。図１０に示される組み合わせ回路においては、入力Ａ１が第１のＸＯＲ群５２へと入力され、入力Ｂ１が第２のＸＯＲ群５４へと入力され、入力Ｂ２が、第３のＸＯＲ群５６へと入力される構成とされている。これらのＸＯＲ群５２、５４、５６が、本発明において採用される変数前処理演算を実行するゲートである。このうちＸＯＲ群５２、５４、５６は、共通した入力Ａ１を処理する構成とされていて、回路規模の縮少になっている。各ＸＯＲ群５２、５４、５６の出力は、それぞれＡＮＤ群５８へと入力され、クロスタームが算出された後、再度下流側のＸＯＲ群６０において剰余演算が実行され、ＸＯＲ群６０ａから出力６２が出力され、ＸＯＲ群６０ｂから出力６４が生成されている。図１０にいては、１つの乗算を行う単位が破線ＢＬで示されており、変数前処理（ＸＯＲ）－クロスターム演算部（ＡＮＤ）－剰余演算部（ＸＯＲ）の３段構成により、１つの乗算器が構成されているのが示されている。図１０に示されるように、各乗算をXOR-AND-XORの構成とし、かつ、共通の入力に対して行うＸＯＲ演算を乗算間で統一すれば、そのＸＯＲ演算回路を乗算回路間で共有できることになる。１個分の乗算に相当する部分（変数前処理演算＊２＋新しいクロスターム構成部＋新しい剰余演算部）の回路規模が通常の乗算回路と同じか若干大きい程度であれば、乗算群全体としての回路規模を減少させることが可能となる。

【００７０】

図１１は、図１０に示した従来の組み合わせ回路を、本発明において採用する３段構成の乗算器を使用して実装化した場合の組み合わせ回路の構成の別の実施の形態を示した図である。図１１に示される組み合わせ回路においては、入力６６、入力６８、入力７０、入力７２が、それぞれ変数前処理演算を実行するＸＯＲ群７４、７６、７７、７８へと入力され、ＸＯＲ群のそれぞれの出力が、クロスターム構成演算を実行するＡＮＤ群８０、８２へと入力されている。

【００７１】

ＡＮＤ群８０、８２の出力は、加算回路８４へと入力されて加算が実行され再度共有された下流側のＸＯＲ群８６において剰余演算が実行されて、乗算が行われ、出力８８が生成される構成とされている。図１１に示した乗算器１単位の構成は、枠Ｂｘで示した内側において形成されている。図１０との相違点は、入力側のＸＯＲ群７４、７６、７７、７８が共有されない場合であっても、本発明においては、下流側、すなわち出力側の剰余演算を実行するＸＯＲ群８６を共有する構成とすることができる。

【００７２】

さらに、本発明においては、入力側、すなわち変数前処理演算を実行するＸＯＲ群７４～７８を共有化し、剰余演算を実行するＸＯＲ群を同時に共有させることにより、さらに全体としての組み合わせ回路の構成を縮小することができる。

【００７３】

図１２は、図９に示した組み合わせ回路を、３段構成の乗算器を使用して本発明の組み合わせ回路とした、さらに別の実施の形態を示した図である。なお、図１２に示した組み合わせ回路は、図６において示したＲＳ符号誤り訂正のための復号回路（ｅ＝２）の処理部１２の一部を構成する組み合わせ回路の実施の形態である。図１２に示した組み合わせ回路においても、図９に示した従来例と同様にＳ０～Ｓ３Ｑが入力されている。図１２に示すように、入力Ｓ０の変数前処理演算を行うＸＯＲゲート９０は、枠で示すように、ＡＮＤゲート９２、９４、９６に対応する３つの乗算器により共有されている。さらに下流側においては、剰余演算部９８についても複数の乗算器により共有されていて、変数前処理演算を行うＸＯＲゲートおよび剰余演算を実行するＸＯＲゲートの双方が共有されているのが示されている。また、図１０、図１１、図１２と同様に、乗算器の１つの単位は、鎖線により示されている。図１２に示した組み合わせ回路においては、回路中の変数前処理演算部は８個、クロスターム構成演算部は７個、剰余演算部は４個、８ビット幅の加算は２個、クロスターム構成演算部と同ビット幅の加算は３個となる。

【００７４】

このことから、表２で示した乗算回路をもとに、図１２に示した組み合わせ回

路で上述した復号回路の一部を構成させる場合については、そのゲート数は、以下の表３のようにまとめることができる。

【００７５】

【表３】

Composite Field Multiplier	416XOR + 336AND 計 752gate
本発明	
a. ３個中３個の部分体乗算を改変	458XOR + 210AND 計 668gate
b. ３個中２個を改変	444XOR + 252AND 計 696gate
c. ３個中１個を改変	430XOR + 294AND 計 724gate

【００７６】

表３ａ～ｃに示されるように、単体の乗算をあえて最小規模にはしなかったにもかかわらず、回路全体としてはより回路規模が縮小されることが見出された。

【００７７】

図１３は、誤り訂正能力ｔ＝２～８の図６に示した誤り訂正回路の処理部１２に使用する場合において必要とされるＸＯＲゲート数と、それに対応する乗算器数とを示した図である。図１３は、縦軸をＸＯＲの全数とし、横軸を乗算器の個数として示されており、この場合には、図６の誤り訂正用の復号回路は、ｍ＝８で、既約多項式が$x^8+x^4+x^3+x^2+1$であるものとして算出した。この場合、乗算を６６２、加算を５３１、２乗演算を３０回使用する。表３および図１３から理解されるように、乗算器単体の乗算をあえて最小規模にはしなかったにもかかわらず、本発明の構成を採用することにより、組み合わせ回路全体としては回路規模をより縮小することができることが示される。

【００７８】

図１３に示した実施の形態においては、具体的な回路構造を明確にする目的から、乗算における変数前処理部、クロスターム構成部、剰余演算部をすべての乗算について同一であるものとして説明した。実装上の段階においては、さらに良好な結果を得るために、どれだけのXORを変数前処理部・剰余演算部に配置するか（つまり，表記a～cのいずれを用いるか）を、回路中の乗算ごとに変更して最

適化することも可能である。さらに、図１３に示した実施の形態においては、回路入力に対する演算数の割合が少なく、体変換のオーバーヘッドもあるので、ゲート数減少の効果はさほど大きくはないものの、実用上は入力に対する演算の割合がかなり高くなるため、ゲート数削減効果による回路規模の削減は、図１４に示すように顕著なものとなる。

【００７９】

セクション３　＜誤り訂正アルゴリズム＞

以下、本発明の復号回路、誤り訂正装置において使用される誤り訂正アルゴリズムについて詳細に説明する。

【００８０】

（従来技術の概要）

Ａ．＜Yule-Walker方程式を解く、または誤り位置多項式を求めるための従来の手法とその問題点＞

本発明においては、ＧＦ(２^{m})上で定義された次の連立一次方程式の解を組み合わせ回路を用いて計算するための効率的なアルゴリズムを見出すことが必要である。

【００８１】

$$\begin{pmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ S_1 & S_2 & \cdots & S_l \\ \vdots & & \ddots & \vdots \\ S_{l-1} & S_l & \cdots & S_{2l-2} \end{pmatrix} \begin{pmatrix} \Lambda_l^{(l)} \\ \vdots \\ \vdots \\ \Lambda_1^{(l)} \end{pmatrix} = \begin{pmatrix} S_l \\ \vdots \\ \vdots \\ S_{2l-1} \end{pmatrix}$$

上記式中、

$$S_0, S_1, \cdots, S_{2t-1}$$

は、与えられたGF(２^{m})の元であり、$\Lambda^{(l)}_i$が未知の量である。

【００８２】

上記の連立一次方程式において、左辺の行列は、右斜め方向(対角線と交わる方向)に同じ成分が並ぶという規則的な構造をしており、Hankel行列といわれる。一般にこのタイプの方程式は、Yule-Walker方程式と呼ばれ、誤り訂正符号の

理論をはじめとして時系列解析や信号処理の分野でも現れるなど、広い応用範囲を有していることが知られている。誤り訂正アルゴリズムにおいては、誤り位置多項式を決定する部分にこのYule-Walker方程式が現れることになる。そこで、本発明においては、上述したYule-Walker方程式の解を得るためのアルゴリズムを、リード・ソロモン符号の復号を行うために使用される誤り訂正アルゴリズムに適用するものである。

【００８３】

上述したYule-Walker方程式の解法としてよく知られているものとして、Levinsonのアルゴリズム、Levinson-Durbinのアルゴリズム等がある。これらのアルゴリズムは、いずれも行列のサイズｌが小さい所から計算を始め、再帰的に行列のサイズが大きい方程式の解を決定していくものである。また、計算量は共にｌ^{2}のオーダである。しかしながら、これらのアルゴリズムは、計算ステップの中に割り算の操作を含んでいる。このことは、アルゴリズムの実行を組み合わせ回路として実装することを考えたとき、分母が０か否かに応じた条件分岐が発生することを意味する。この条件分岐が発生することにより、条件分岐の各々に対して別々の回路を用意しなくてはならないので必要とされる回路サイズは 行列のサイズが大きくなるにしたがって組み合わせ的な速さで大きくなってしまうという本質的な問題を生じる。

【００８４】

また、本発明においては、Yule-Walker方程式の解を求めることによって、特にリード・ソロモン符号の復号化において、誤り位置多項式を決定することを目的とする。従来Yule-Walker方程式の解法として用いられている方法としては、Peterson法、Berlekamp-Massey法、Euclid法などを挙げることができる。これらはいずれも訂正可能な誤りの最大数ｔに関して、多項式オーダの計算量で誤り位置多項式の係数を計算するものである。しかしながら、Berlekamp-Massey法とEuclid法を組み合わせ回路で表現と以下に述べる問題が生じる。

【００８５】

まず、Berlekamp-Massey法に関しては、アルゴリズムの中にやはり複数個の条件分岐を含むことになることが不可避である。したがって、これを組み合わせ回

路に展開するときには上述した理由と同じ理由で回路サイズは組み合わせ的に増大する。一方、Euclid法においては多項式の乗除算がアルゴリズムの基本となるが、割り算の分母に現れる多項式の次数が前もってわからないために、ここに条件分岐の入る余地がある。また、この条件分岐に起因してBerlekamp-Massey法の場合と同様の組み合わせ的な回路規模の増大を生じさせてしまうことになる。

【００８６】

Ｂ．＜ 組み合わせ回路に適したYule-Walker方程式および誤り位置多項式の計算法の方針＞

上述したように、Levinson(-Durbin)法、Berlekamp-Massey法、Euclid法は共に条件分岐を含むため、組み合わせ回路化という観点からは問題がある。そこで、Yule-Walker方程式の解法を組み合わせ回路で実現するためには場合分けのないアルゴリズムを見出すことが必要であり、これが、本発明のアルゴリズムにおける本質的な方針として与えられる。

【００８７】

この場合、上述のアルゴリズムとして、リード・ソロモン符号の復号化で知られているPeterson法を用いることができる。Peterson法のアプローチはYule-Walker方程式を直接解くことになる。この際、Yule-Walker方程式の解は次のようにCramerの公式を用いて行列式の形で表示することができる。

【００８８】

$$\Lambda_i^{(l)} = \frac{\tilde{\Lambda}_i^{(l)}}{\tilde{\Lambda}_0^{(l)}}, \quad i = 1, \cdots, \ell$$

$$\tilde{\Lambda}_0^{(l)} = \begin{vmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ S_1 & S_2 & \cdots & S_l \\ \vdots & \vdots & \ddots & \vdots \\ S_{l-1} & S_l & \cdots & S_{2l-2} \end{vmatrix}$$

$$\tilde{\Lambda}_i^{(l)} = \begin{vmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ \vdots & \cdots & \ddots & \vdots \\ S_{l-i-1} & S_{l-i} & \cdots & S_{2l-i-2} \\ S_{l-i+1} & S_{l-i+2} & \cdots & S_{2l-i} \\ \vdots & \cdots & \ddots & \vdots \\ S_l & S_{l+1} & \cdots & S_{2l-1} \end{vmatrix}, \quad i = 1, \cdots, l-1$$

$$\tilde{\Lambda}_l^{(l)} = \begin{vmatrix} S_1 & S_2 & \cdots & S_l \\ S_2 & S_3 & \cdots & S_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_l & S_{l+1} & \cdots & S_{2l-1} \end{vmatrix}.$$

したがって、それぞれの

$$l = 1, \cdots, t$$

に対して、行列式$\Lambda^{(1)}{}_0$を求め、誤りの個数ｅに対して、行列式、

$$\tilde{\Lambda}_i^{(e)}, i = 1, \cdots, e$$

を計算すればよいことになる。

【００８９】

しかしながら、行列式の展開をそのまま回路として実現すると、ｔが増大するにしたがって必要となる乗算器の個数が飛躍的に増大するので同様に直接的な適用は困難である。このため、本発明においては、Hankel行列の帰納的な構造を利用して計算量の削減を行う。このためのアプローチとしてKatayama-Moriokaによる$\Lambda^{hat(1)}{}_i$の計算と従来の方法と対比して説明する。

【００９０】

まず、Katayama-Moriokaの中での$\Lambda^{(1)}{}_i$の計算アルゴリズムをｌ＝１からｌ＝４まで書き下すと図１４に示す形態となる。

【００９１】

次にHankel行列の行列式を帰納的なアプローチで計算する別な手法として、Kogaによる方法を対比のために説明する。Kogaによる方法によれば、まず、次の新たな誤り位置多項式、

$$i;i+2u D(X,Y)$$

が定義されている。ここで、

$$i;i+2u D(X,Y)=\begin{pmatrix} S_i+YX^i & S_{i+1}+YX^{i+1} & \cdots & S_{i+u}+YX^{i+u} \\ S_{i+1}+YX^{i+1} & S_{i+2}+YX^{i+2} & \cdots & S_{i+u+1}+YX^{i+u+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i+u}+YX^{i+u} & S_{i+u+1}+YX^{i+u+1} & \cdots & S_{i+2u}+YX^{i+2u} \end{pmatrix}.$$

である。

【００９２】

そして、この新たな誤り位置多項式を計算するため、ｉ番目のシンドロームＳ$_i$を(1、1)成分に持つようなHankel行列

$$i;i+2u Q$$

を考え、

$$i;i+2u Q$$

から複数の行と列を対称に抜き去った行列式をQ行列式と定義する。Q行列式は、一般に対角成分に現れるシンドロームの添え字番号を左上から順に指定してやれば唯一に定まるものである。ここで、Q行列式を添え字の列、

$$[a_1,a_2,\cdots,a_p]$$

で表すことができる。Kogaによる方法では、Q行列式を用いて誤り位置多項式、

$$i;i+2u\ D\ (X\ ,\ Y\ )$$

を計算するアルゴリズムが提示されている。

【００９３】

上述した従来手法は、いずれにしても次に述べるような問題点を抱えている．まず、従来例１の中で用いられた$\Lambda^{(1)}_i$の計算アルゴリズムに関しては計算すべき行列式の非対称性に起因して展開の右辺に次々と新しい項が出現し、その

結果として行列のサイズが大きくなるにしたがって必要とされる乗算器の個数は組み合わせ的に増大することになる。そこで、このような組み合わせ的発散の程度がなるべく小さいアルゴリズムが望ましい。

【００９４】

次にKogaのアルゴリズムについてであるが、Kogaの定義したＱ行列式は対称な行列式であり、このことによって乗算器数の削減が実現されている。しかしながら、Kogaのアルゴリズムは、最小距離が偶数の場合のBCH符号もしくはリード・ソロモン符号でしか適用することができないという制限がある。Kogaによれば、この制限を緩和できる場合もあることを開示しているものの、緩和できる例としては、単にbinary narrow sense BCH符号の場合に限定されてしまうことになる。

【００９５】

本発明においては、光通信システムへの適用を行うため、(255、239)リード・ソロモン符号(最小距離=１７)の復号化を組み合わせ回路で効率よく実現することが方針として要求されることになる。このため、何らかの手法を用いて最小距離の偶数、奇数にかかわらず効率的な計算を実行させることができるアルゴリズムが必要となる。

【００９６】

Ｃ．＜本発明において使用される用語の定義＞

以下に、本発明のアルゴリズムを詳細に説明する前に、本発明において使用する各タームを明確にするために説明を行う。

（１）シンドローム

一般に、ガロア拡大体ＧＦ（2^m）の原始元をａとし、ｈ＜2^m－１を正整数とするとき、

$$G(x) = (x-1)(x-a)(x-a^2)\cdots(x-a^{h-1})$$

を生成多項式とする、符号長がｎ＝2^m－１の2^m元巡回符号をＧＦ（2^m）上のリード・ソロモン符号として定義する。すなわち、ｋ＝ｎ－ｈとしｋ個の送信記号を係数にもつｋ次多項式をＭ（ｘ）とするとき、Ｍ（ｘ）にx^{n-k}を乗算

し、その結果を次のようにＧ（ｘ）で除算し、剰余Ｒ（ｘ）を算出する。

【００９７】

$$M(x)x^{n-k} = Q(x)G(x) + R(x)$$

ついで、長さｎの符号化された系列を係数に持つ多項式(送信多項式)を

$$W(x) = M(x)x^{n-k} - R(x) = Q(x)G(x)$$

で定義する。このとき、符号化された送信系列は左端に ｋ 個の情報記号を持ち、それらにｈ＝ｎ－ｋ個の検査記号が続く組織符号の形になっている。リード・ソロモン符号の最小距離d_{min}は、$d_{min}=h+1$で与えられる。また、訂正可能な誤りの最大個数ｔはｔ＝[ｈ／２]で与えられる。

【００９８】

一方、受信された系列よりもとの送信系列を推定する復号化のアルゴリズムは以下のように与えられる。

（２）　シンドロームの計算と誤りの検出

ここで、１個の誤りが生じたとし、それらの位置を$i_0, \cdots, i_{l-1}$、誤りの値を $E_{i0}, \cdots, E_{il-1}$ とする。$E_{i0}, \cdots, E_{il-1}$を係数に持つ多項式を、

$$E(x) = E_{i_0}x^{i_0} + \cdots + E_{i_{l-1}}x^{i_{l-1}}$$

で定義すると、受信系列b_0、．．．、b_{n-1}を係数とする多項式は、

$$Y(x) = \sum_{i=0}^{n-1} b_i x^i = W(x) + E(x)$$

で与えられる。Ｙ（ｘ）を受信多項式と定義する。

次に、受信多項式Ｙ（ｘ）からシンドローム、

$$S_i = Y(\alpha^i) \in GF(2^m), \quad i = 0, 1, \cdots, 2t-1$$

を計算する。ここで、

$$W(a^i) = 0, i = 0, 1, \cdots, 2t-1$$

なので、シンドロームは

$$S_i = E(a^i), \quad i = 0, 1, \cdots, 2t-1$$

を満たす。したがって、誤りがなければシンドロームはすべて0となり、シンドロームの値によって誤りの有無が判定できることになる。

【００９９】

（３）誤りの個数と位置の特定

発生した誤りの個数を仮に１個であると仮定し、発生した位置を、

$$i_0, \cdots, i_{l-1}$$

と仮定する。すなわち、

$$b_{i_0}, \cdots, b_{i_{l-1}}$$

の値が誤っていると仮定する。誤りの個数ｌと、下記式の誤り位置、

$$i_0, \cdots, i_{l-1}$$

を特定するために、

$$a^{-i_0}, \cdots, a^{-i_{l-1}}$$

を根として持つ下記多項式、

$$\Lambda^{(l)}(x) = \Pi_{k=0}^{l-1}(1 - a^{i_k}x) = 1 + \Lambda_1^{(l)}x + \cdots + \Lambda_{l-1}^{(l)}x^{l-1} + \Lambda_l^{(l)}x^l$$

を定義する。上記式中、

$$a^{-i_0}, \cdots, a^{-i_{l-1}}$$

を誤りロケータ、$\Lambda^{(l)}(x)$を誤り位置多項式という。

【０１００】

さらに、

$$\Lambda_1^{(l)}, \cdots, \Lambda_l^{(l)}$$

は、誤り位置多項式をｘに関して展開したときの展開係数であり、

$$a^{i_0}, \cdots, a^{i_{l-1}}$$

の基本対称式で与えられる。

【０１０１】

ここで、下記式

$$\Lambda_1^{(l)}, \cdots, \Lambda_l^{(l)}$$

は、次の連立一次方程式、

$$\begin{pmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ S_1 & S_2 & \cdots & S_l \\ \vdots & & \ddots & \vdots \\ S_{l-1} & S_l & \cdots & S_{2l-2} \end{pmatrix} \begin{pmatrix} \Lambda_l^{(l)} \\ \vdots \\ \vdots \\ \Lambda_1^{(l)} \end{pmatrix} = \begin{pmatrix} S_l \\ \vdots \\ \vdots \\ S_{2l-1} \end{pmatrix}$$

を満たす。これはＡ．で説明したYule-Walker方程式に他ならない。この段階では、ｌは未知であるが、実際に生じた誤りの個数が１≦ｅ≦ｔであるとき、左辺のHankel行列は、ｌ＝ｅの場合は正則であって、ｔ≧ｌ＞ｅの場合は非正則となることが知られている。したがって、ｌ＝１，．．．，ｔに対して左辺のHankel行列の行列式を計算し、値が非ゼロの最大の整数をもって誤りの個数ｅと定めればよい。そしてｌ＝ｅの場合にこの方程式を解くことにより、誤り位置多項式を求めることができる。

【０１０２】

本発明において、誤り位置を特定するためには、誤りロケータ、すなわち、誤り多項式$\Lambda^{(e)}$（ｘ）＝０の根を求めればよい。このために下記式

$$a^{-i}, i = 0, 1, \cdots, n-1$$

を逐次代入して実際に誤り位置多項式の零点となるか否かを調べるという手法を取ることができる。この手法をChien探索という。誤り位置多項式の零点を、

$$a^{-i_0}, \cdots, a^{-i_{e-1}}$$

とするとき、$i_0, \ldots, i_{e-1}$が実際の誤り位置を与える。

【０１０３】

（４）誤り値の計算

誤り値の算出は、下記式で示されるVandermonde型の連立線形方程式、

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ a^{i_0} & a^{i_1} & \cdots & a^{i_{e-1}} \\ \vdots & \vdots & \ddots & \vdots \\ a^{i_0(e-1)} & a^{i_1(e-1)} & \cdots & a^{i_{e-1}(e-1)} \end{pmatrix} \begin{pmatrix} E_{i_0} \\ \vdots \\ \vdots \\ E_{i_{e-1}} \end{pmatrix} = \begin{pmatrix} S_0 \\ \vdots \\ \vdots \\ S_{e-1} \end{pmatrix}.$$

を解くことによって得られる。シンドロームを係数とする多項式Ｓ（ｘ）を、

$$S(x) = S_0 + S_1 x + S_{2t-1} x^{2t-1}$$

とおく。さらに、

$$\Omega(x) = \Lambda^{(e)}(x) S(x) \mod x^{2t-1}$$

とおく。Ω（ｘ）を誤り評価多項式という。この場合に、Vandermonde型の連立線形方程式の解は、

$$E_{i_k} = \frac{\Omega(a^{-i_k})}{\Lambda'(a^{-i_k})}, \quad i = 0, \cdots, e-1$$

によって求めることができる。これをForneyのアルゴリズムという。したがって、誤り位置と誤り値とがわかれば、それらを受信した入力ディジタル信号から減算することにより、誤りの訂正されたディジタル信号出力を得ることができる。

【０１０４】

Ｄ．本発明のYule-Walker方程式解法アルゴリズム

我々の課題はＧＦ（2^m）上で定義された次のYule-Walker方程式の解を組み合わせ回路を用いて計算するための効率的なアルゴリズムを見出すことである．

$$\begin{pmatrix} S_0 & \cdots & S_{l-1} \\ \vdots & \ddots & \vdots \\ S_{l-1} & \cdots & S_{2l-2} \end{pmatrix}\begin{pmatrix} \Lambda_l^{(l)} \\ \vdots \\ \Lambda_1^{(l)} \end{pmatrix}=\begin{pmatrix} S_l \\ \vdots \\ S_{2l-1} \end{pmatrix}.$$

ここで、

$$S_0, S_1, \cdots, S_{2t-1}$$

は、与えられたＧＦ（2^m）の元であり、$\Lambda_i^{(l)}$が未知の量である。

【０１０５】

本発明においては、このYule-Walker方程式の解をCramerの公式を用いて、図１５のように行列式の形で表示し、その帰納的な構造を利用して行列式の効率のよい計算法を求めるものである。

【０１０６】

図１５に示した行列式を計算するために、本発明においては、次のJacobiの公式に注目する。

＜Jacobiの公式＞

Ａ＝（a_{ij}）を単位元１をもつ可換環上のｎ次正方行列とする．Ａの（ｉ，ｊ）余因子をΔ_{ij}とする。添え字の集合、

$$\mu = \{i_1, \cdots, i_r, (i_1 < \cdots < i_r)\}, \nu = \{j_1, \cdots, j_r, (j_1 < \cdots < j_r)\}$$

に対する小行列式$A_{\mu\nu}^{(r)}$の余因子を$\Delta_{\mu\nu}^{(r)}$とするとき、下記式が成り立つ。

【０１０７】

$$\begin{vmatrix} \Delta_{i_1 j_1} & \cdots & \Delta_{i_1 j_r} \\ \vdots & \ddots & \vdots \\ \Delta_{i_r j_1} & \cdots & \Delta_{i_r j_r} \end{vmatrix} = (\det A)^{r-1}\Delta_{\mu\nu}^{(n-r)}.$$

本発明では特に、

$$\mu = \{i_1, i_2\}, \nu = \{j_1, j_2\}$$

とした場合に成り立つ下記式、

$$\Delta_{i_1 j_1} \cdot \Delta_{i_2 j_2} - \Delta_{i_2 j_1} \cdot \Delta_{i_1 j_2} = (\det A)\Delta_{\mu\nu}^{(n-2)}$$

を用いることができる。

【０１０８】

このJacobiの公式を用いて$\Lambda^{hat}_{i}(l)$を計算するために次のように考える。

まず、$\Lambda^{hat}_{i}(l+1)$は下記式の形をしている。

【０１０９】

$$\tilde{\Lambda}_0^{(l+1)} = \begin{vmatrix} S_0 & S_1 & \cdots & S_{l-1} & S_l \\ S_1 & \ddots & & \vdots & S_{l+1} \\ \vdots & & \ddots & \vdots & \vdots \\ S_{l-1} & \cdots & \cdots & S_{2l-2} & S_{2l-1} \\ S_l & S_{l+1} & \cdots & S_{2l-1} & S_{2l} \end{vmatrix}.$$

この行列式をよく眺めてみると、$\Lambda^{hat}_{i}(l)$は、$\Lambda^{hat}_{i}(l+1)$から（l＋1－i）行および第１列を取り除いたものであり、$\Lambda^{hat}_{0}(l)$は$\Lambda^{hat}_{0}(l+1)$から第１行および第１列を取り除いたものことがわかる。つまり、$\Lambda^{hat}_{0}(l)$は、$\Lambda^{hat}_{i}(l)$が、それぞれ$\Lambda_{i}(l+1)$の（l＋1，l＋1），（l＋1，l＋1－i）余因子であるため、Jacobiの公式において、

$$i_1 = j_1 = l + 1 - i, i_2 = j_2 = l + 1.$$

として、

$$\Delta_{l+1,l+1} = \tilde{\Lambda}_0^{(l)}, \quad \Delta_{l+1,l+1-i} = \Delta_{l+1-i,l+1} = \tilde{\Lambda}_i^{(l)}$$

とおく。さらに、$\Lambda^{hat}_{0}(l+1)$の（l＋1－i，l＋1－i）余因子を$\Gamma_{i}(l+1)$と定義する。ここで、$\Gamma_{i}(l+1)$の構成を図１６に示す。

そして、Jacobiの公式を用いることにより、

$$\Gamma_i^{(l+1)} \tilde{\Lambda}_0^{(l)} + (\tilde{\Lambda}_i^{(l)})^2 = \tilde{\Lambda}_0^{(l+1)} \Gamma_{i-1}^{(l)}, \quad i = 1, \cdots, l$$

を得る。

【０１１０】

この公式を用いると、Λ$^{hat}_{i}$（１）の計算は対称行列の行列式であるΓ$_{i}$（１＋１）の計算に帰着される。ただし、Λ$^{hat}_{i}$（１）を求めるには、Γ$_{i}$（１＋１）の計算に加えて、２×１個の乗算と１個の平方根をとる操作が必要となる。平方根を取る計算と２乗を行う計算とは、線形演算として実現できるため加算とほぼ同等のコストで回路として実現可能である。したがって、これらは非線形演算回路である乗算器に比べて非常に小さなコストしか要しない。そこで我々は乗算器にのみ注目し、その個数を問題とする。提案するアルゴリズムの場合、ＧＦ（２^{m}）は標数が２であること、およびΓ$_{i}$（１）はすべて対称あることから、行列式の余因子展開において対角線に関して非対称な配置より生じる項は必ずキャンセルする．例えば、３×３の対称行列を例にとって余因子展開を計算してみると、

$$\begin{vmatrix} a & b & c \\ b & d & e \\ c & e & f \end{vmatrix} = adf + ae^2 + b^2f + bec + c^2d + bec = adf + ae^2 + b^2f + c^2d$$

となって、対角線に関して非対称な配置より生じる項ｂｅｃは対称性により必ず２度出てくるために、キャンセルされる。このため、本発明のアルゴリズムを乗算器を含む組み合わせ回路に使用した場合には、要求される乗算器の個数が低減できることになる。

【０１１１】

ここで、

$$\Gamma_i^{(l)}, l = 1, 2, \cdots, t+1, i = 0, 1, \cdots, t$$

を帰納的に計算するアルゴリズムの一般形は以下のように与えられる．

0.　Γ$_{1}$$^{(1)}$＝１，Γ$_{0}$$^{(2)}$＝Ｓ$_{0}$，Γ$_{1}$$^{(2)}$ Ｓ$_{2}$

1.　１＞２，ｉ＝１のとき

$$\Gamma_0^{(l)} = S_{2l-4}\Gamma_0^{(l-1)} + \sum_{k=1}^{l-2} S_{2l-4-k}^2 \Gamma_{k-1}^{(l-2)}.$$

2.　１＞２，ｉ＝１...，１－１のとき

まず、アルゴリズムを記述する際の補助的な量を一つ定義する。

【０１１２】

$$\{i_1, \cdots, i_n\}$$

を、添え字の集合とするとき、ｄｅｔ[{ｉ１，．．．，ｉｎ}]を

$$\det[\{i_1, \cdots, i_n\}] = \begin{vmatrix} S_{i_1} & S_{i_2} & \cdots & S_{i_n} \\ S_{i_2} & S_{2i_2-i_1} & \cdots & S_{i_2+i_n-i_1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i_n} & S_{i_2+i_n-i_1} & \cdots & S_{2i_n-i_1} \end{vmatrix}$$

と定義する。正確にいうと、ｄｅｔ[{ｉ１，．．．，ｉｎ}]は、第1行目が、

$$S_{i_1}, \cdots, S_{i_n}$$

であって、（ｐ、ｑ）成分が下記式、

$$S_{i_p+i_q-i_1}$$

である対称行列の行列式であり、Hankel行列式$\Lambda_0^{(l)}$からいくつかの行と列とを対称に抜き去って得られる行列式である。ここで、ｄｅｔ[{ｉ１，．．．，ｉｎ}]を用いると$\Gamma_i^{(l)}$は次のように計算される。

【０１１３】

$$\Gamma_i^{(l)} = S_{2l-2}\Gamma_{i-1}^{(l-1)} + \sum_{k=1,k\neq i}^{l-1} S_{2l-2-k}^2 \det[\{0,1,\cdots,l-2\} - \{l-1-i, l-1-k\}]..$$

上記式中、ｄｅｔ[{０，１，．．．，ｌ－２}－{ｌ－１－ｉ，ｌ－１－ｋ}]は、$\Gamma_i^{(l-1)}$から対称に２つのｌ－１－ｉ，ｌ－１－ｋ行と、２つのｌ－１－ｉ，ｌ－１－ｋ列とを抜き去って得られる対称行列の行列式であって、これはｋ＝１の場合とｉ＝１との場合には、それぞれ、下記式、

$$\Gamma_{i-2}^{(l-2)}, \Gamma_{k-2}^{(l-2)}$$

に一致することに注意されたい。

【０１１４】

3. 一般にｄｅｔ[{ｉ$_1$，．．．，ｉ$_n$}]は、次のように計算される。

【０１１５】

$$\det[\{i_1,\cdots,i_n\}] = S_{2i_n-i_1}\det[\{i_1,\cdots,i_{n-1}\}] + \sum_{k=1}^{n-1} S^2_{i_n-i_1+i_k}\det[\{i_1,\cdots,i_{n-1}\}-\{i_k\}].$$

Ｅ．＜本発明のアルゴリズムのリード・ソロモン符号の復号化への適用＞

以下に、Ｄ．で述べたYule-Walker方程式の本発明の解法アルゴリズムを、リード・ソロモン符号へと応用した場合についての実施の形態を説明する。Yule-Walker方程式自身は、通常は、次元(未知数の個数)は一定のものとして与えられるのであるが、リード・ソロモン符号の復号化の場合、次元(誤りの個数に対応している)も未知なので、これも含めて決定しなければならないことになる。

【０１１６】

（１）$\Gamma_i^{(1)}$の計算

シンドロームの系列、

$$S_0, S_1, \cdots, S_{2t-1}$$

が与えられたとき、Ｄ．において説明したアルゴリズムに従い、

$$\Gamma_i^{(l)}, l = 1,2,\cdots,t+1, i = 0,\cdots,t$$

を計算する。この計算の過程で、

$$\tilde{\Lambda}_0^{(l)} = \Gamma_0^{(l+1)}, l = 1, \cdots, t$$

も、同時に計算される。なお、本発明者らは、リード・ソロモン符号の高速復号化を行うために、組み合わせ回路としての実現を念頭に置いているが、本発明においては組み合わせ回路と共に使用することに限定されるものではなく、本発明の誤り訂正アルゴリズムは、順序回路を用いて誤り訂正装置として実装することも可能である。

【０１１７】

（２）誤りの個数の決定

実際に生じた誤りの個数を、ｅで表す。ここで、ｅは

$$\tilde{\Lambda}_0^{(l)} = \Gamma_0^{(l+1)}, l = 1, \cdots, t$$

の値から、$\Lambda^{hat}{}_0(1) \neq 0$を満たす最大の１として求めることができる。

【０１１８】

（３）誤り位置多項式の決定

誤りの個数を決定した結果、もしｅ＜ｔが判明した場合には、$\Lambda^{hat}{}_0(e+1) = 0$なので、本発明のアルゴリズムに従い、

$$\tilde{\Lambda}_i^{(e)} = \sqrt{\Gamma_i^{(e+1)} \tilde{\Lambda}_0^{(e)}}, \ i = 1, \cdots, e.$$

のように簡単化することができる。誤りロケータは、誤り位置多項式の零点であるので、誤り位置多項式の係数の定数倍によって不変である。したがって、$\Lambda^{hat}{}_i(e)$の代わりに、下記式

$$\sqrt{\Gamma_i^{(e+1)}}$$

を用いることができる。つまり、上式に現れている掛け算は必要ない。一方、ｅ＝ｔであることが判明した場合には、

$$\tilde{\Lambda}_{b_i}^{(e)} = \sqrt{\Gamma_i^{(e+1)} \tilde{\Lambda}_{b_0}^{(e)} + \tilde{\Lambda}_{b_0}^{(e+1)} \Gamma_{i-1}^{(e)}}, \quad i = 1, \cdots, e.$$

に従って誤り位置多項式を計算することになる。このとき、我々のアルゴリズムでは、最小距離が奇数（＝２ｔ＋１）の場合には、計算できないS_{2t}が見かけ上必要になるように見える。しかし、本発明の式は、シンドロームに関する恒等式であるため、S_{2t}に関する恒等式にもなっている。そして、計算すべき$\Lambda^{hat}{}_i(t)$は、シンドロームS_{2t}を含まないので、$\Lambda^{hat}{}_0(t+1)$と、$\Gamma_i^{(t+1)}$の余因子展開に現れるS_{2t}は必ずキャンセルする。具体的にいうと、$\Gamma_i^{(t+1)}$を余因子展開したときに現れる項のうち、S_{2t}を含むものは、$\Gamma_{i-1}^{(t)} S_{2t}$であるから、$\Gamma_i^{(t+1)} \Lambda^{hat}{}_0(t)$を展開したとき、$S_{2t}$を含む項は$\Lambda^{hat}{}_0(t) \Gamma_{i-1}^{(t)} S_{2t}$である。一方、$\Lambda^{hat}{}_0(t+1)$を余因子展開したときに現れる項のうち、$S_{2t}$

を含むものは、$\Lambda^{hat}{}_0{}^{(t)} S_{2t}$であるから、$\Lambda_0{}^{(t+1)} \Gamma_{i-1}{}^{(t)}$を展開したとき$S_{2t}$を含む項は、$\Lambda^{hat}{}_0{}^{(t)} \Gamma_{i-1}{}^{(t)} S_{2t}$である。したがって、両者は必ずキャンセルする。

【０１１９】

こうして、$\Lambda_0{}^{(t+1)}$と、$\Gamma_i{}^{(t+1)}$を余因子展開したときに現れる項のうち、S_{2t}を係数にもつ項は計算する必要がないことが示される。このようにして、本発明のアルゴリズムは、任意の最小距離をもつリード・ソロモン符号に適用することができることとなる。同時に、乗算器数の削減という観点からも、S_{2t}を含む項の乗算が必要ないので、Kogaアルゴリズムと比較しても本発明のアルゴリズムは、優位になる。また、上述したように、平方根を算出する計算は、加算とほぼ同等のコストで回路として実現可能であり、これは乗算器に比べて非常に小さなコストしか要しないものである。

【０１２０】

Ｆ．＜リード・ソロモン符号の復号化への適用例＞

上述したＥにおいて説明した本発明の誤り訂正アルゴリズムをｔ＝４のリード・ソロモン符号の復号化へ適用する実施の形態について、以下に説明する。ｔ＝４の場合には、本発明により、以下の各式が決定される。ただし、簡単のため、

$$\det[\{i_1, \cdots, i_n\}] = \det i_1 \cdots i_n$$

として表す。

（１）$\Gamma_i{}^{(1)}$，ｉ＝０，．．．，１－１，．．．，５の計算

本発明による計算結果を、図１７に示す。

【０１２１】

（２）誤りの個数の決定

８１）で算出された$\Gamma_i{}^{(1)}$によって、

$$\tilde{\Lambda}_0^{(l)} = \Gamma_0^{(l+1)}, l = 1, \cdots, 4$$

が求まるので、

$$\tilde{\Lambda}_0^{(l)} \neq 0$$

を満たす最大の１，１＝１，２，３，４として誤りの個数ｅが決定できる。

【０１２２】

（３）誤り位置多項式の決定

（２）よる計算により、例えばｅ＝２であることが判明した場合には、誤りロケータ、

$$a^{i_0}, a^{i_1}$$

は、次の代数方程式、

$$\sqrt{\tilde{\Lambda}_0^{(2)}} + \sqrt{\Gamma_1^{(3)}}\, x + \sqrt{\Gamma_2^{(3)}}\, x^2 = 0$$

を解くことによって求められる。一方、ｅ＝４であることが判明した場合には、上述したように、

$$\tilde{\Lambda}_i^{(4)} = \sqrt{\Gamma_i^{(5)} \tilde{\Lambda}_0^{(4)} + \tilde{\Lambda}_0^{(5)} \Gamma_{i-1}^{(4)}}, \quad i = 1, 2, 3, 4$$

によって求めることができる。ただし、

$$\Gamma_i^{(5)}, \Gamma_0^{(6)} = \tilde{\Lambda}_0^{(5)}$$

の計算において、シンドロームＳ$_8$を含む項の計算は上述したように不要である。

【０１２３】

図１８は、上述した本発明の誤り訂正アルゴリズムの概略的なフローチャートを示した図である。本発明の誤り訂正アルゴリズムにおいては、まず、ステップ２００において、シンドロームＳ$_0$，．．．，Ｓ$_{2t-1}$が入力され、ステップ２０１において、誤り多項式Γが算出される。Γ$_0^{(2)}$，．．．．，Γ$_0^{(t+1)}$が求められた段階で、ステップ２０２において、Λ$^{hat}{}_0^{(m)}$＝Γ$_0^{(m+1)}$≠０を満たす最大の整数ｍ、として誤りの個数を決定する。ステップ２０３においては、誤りの個数ｅが、最大の誤りの数に等しいか否かが判断され、

ｅ＝ｔの場合（ｙｅｓ）には、$\Gamma_0^{(e+1)}=\Lambda_0^{hat(e)}$，．．．，$\Gamma_e^{(e+1)}$、$\Gamma_0^{(e+2)}=\Lambda_0^{hat(e+1)}$として、ステップ２０４において、誤り値を算出する。また、ｅ≠ｔの場合（ｎｏ）には、$\Gamma_0^{(e+1)}=\Lambda_0^{hat(e)}$，．．，$\Gamma_e^{(e+1)}$のみを使用して、ステップ２０５において誤り値を計算し、ステップ２０６において、$\Lambda_0^{hat(e)}$，．．．．，$\Lambda_e^{hat(e)}$を得る。

【０１２４】

Ｇ．本発明のアルゴリズムを誤り位置多項式の計算へ適用した場合の計算回路

図１９に本発明で提案するアルゴリズムに基づいた誤り位置多項式の計算回路のブロック図を示す。図２０に示される本発明のアルゴリズムを使用した誤り位置多項式の計算回路は、概ね$\{\Gamma_i^{(m)}\}$計算ブロック１００と、誤りの個数を計算する回路ブロック１０２と、誤り位置多項式の決定を行う回路ブロック１０４とを含んで構成されている。

【０１２５】

図１９の各ブロックの機能を説明すると、回路ブロック１００には、例えば順序回路を使用して入力ディジタル信号から算出されたシンドロームのシリーズが入力される。回路ブロック１００においては、これらのシンドロームから、

$$\Gamma_i^{(m)}, m=1,2,\cdots,t+1, i=0,\cdots,t$$

が、本発明のアルゴリズムにしたがって帰納的に計算される。これはアルゴリズムの詳細の（１）に対応する。

【０１２６】

次に、回路ブロック１０２においては、計算された、

$$\Gamma_0^{(m)}, m=1,2,\cdots,t+1$$

の値から、誤りの個数ｅを算出し、ｅの値にそれぞれ対応する、

$$\Gamma_i^{(e+1)}, i=0,\cdots,e$$

を出力する。ｅ＝ｔの場合には、これらに加えて、さらに、

$$\Gamma_0^{(t+2)} = \tilde{\Lambda}_0^{(t+1)}$$

も出力される。これはアルゴリズムの詳細の（２）に対応する。回路ブロック１０４では、

$$\Gamma_i^{(e+1)}, i = 0, \cdots, e$$

の値を用いて、誤り位置多項式の係数の計算を実行する。これはアルゴリズムの詳細の（３）に対応するプロセスに従って実行される。

【０１２７】

なお、本発明においては、リード・ソロモン符号の高速復号化を行うために、組み合わせ回路としての実現を念頭に置いているが、提案するアルゴリズムを回路サイズの縮小を目的として順序回路を用いて実現することも可能である。

【０１２８】

Ｈ．＜リード・ソロモン符号の復号化に適用した場合の回路サイズ＞

本発明のアルゴリズムを、リード・ソロモン符号の復号化へ適用した場合の、回路サイズについて以下に説明する。上述したとおり、平方根を算出する計算と２乗を行う計算とは、加算とほぼ同等のコストで回路として実現可能であり、これらは乗算器に比べて非常に小さなコストしか要しない。そこで我々は乗算器にのみ注目し、その個数を検討する。

【０１２９】

表４は、本発明のアルゴリズムによって必要とされる乗算器の個数をｔ＝１からｔ＝８までの範囲で示したものである。この図には比較のため、従来例１および従来例２の計算アルゴリズムによる乗算器数も合わせて記載した。

【０１３０】

【表４】

乗算器数の比較

訂正可能な誤りの個数t	1	2	3	4	5	6	7	8
従来例1のアルゴリズム	0	3	17	48	117*	255*	548*	1111*
従来例2のアルゴリズム	2	9	22	49	98	189	351	640
本発明のアルゴリズム	2	7	21	46	94	179	331	597

注 * 印のついた数値は推定値

【０１３１】

表４に示されるように、今回提案するアルゴリズムは、乗算器の個数の観点からみてKogaにより提案されたアルゴリズム（従来例２）よりもすべてのｔで優れていることが示されている。また、光通信分野への応用においては、特に(255、239)リード・ソロモン符号（ｔ＝８）の復号化が重要であるが、これは最小距離が奇数（＝１７）であるためにKogaアルゴリズムは適用できない。しかしながら、本発明のアルゴリズムは、任意の最小距離のリード・ソロモン符号に適用可能であるため、(255、239)リード・ソロモン符号にも用いることができる。これを、表５に示す。

【０１３２】

【表５】

Kogaアルゴリズムと本発明のアルゴリズムの適用範囲の比較

符号の最小距離	・・・	15	16	17	18	・・・
Koga アルゴリズム	・・・	×	○	×	○	・・・
提案するアルゴリズム	・・・	○	○	○	○	・・・

（×はアルゴリズムが適用できないことを表し、○はできることを意味する。Kogaアルゴリズムが偶数の最小距離の場合しか適用できないのに対し、提案するアルゴリズムは任意の最小距離で適用可能．ITUで標準化された(255、239)リード・ソロモン符号は最小距離が１７の場合である。）

【０１３３】

一方、従来例１（Katayama-Morioka）における計算アルゴリズムも任意の最小距離のリード・ソロモン符号に適用可能であるが、乗算器の個数の観点から比較すると今回提案するアルゴリズムは、ｔが4以上では、従来例１のアルゴリズムよりもより少ない数の乗算器しか必要としない。特に、ｔ＝８の場合には、本発明のアルゴリズムは、約４０％の乗算器の削減を実現することが可能となることが示された。具体的な回路サイズで比較すると、ｔ＝８のとき、誤り値の計算に要するゲート数は、現在約１０Ｋゲートであるのに対して、従来例１では、約８０Ｋゲートを要するものと考えられる。しかしながら、本発明におけるアルゴリズムを用いると、誤り多項式の計算を約４０Ｋゲートまで削減することが可能となる。

【０１３４】

図２０には、本発明の誤り訂正装置の概略ブロック図を示す。図２０に示された誤り訂正装置は、入力ディジタル信号を受信して符号化する符号化ブロック１１０と、符号化された入力ディジタル信号ＩＤが入力され、シンドロームを算出するための入力ブロック１１２と、復号回路を含む処理ブロック１１４と、誤り位置および誤り値との出力を使用して誤りが訂正された出力ディジタル信号ＯＤを出力する出力ブロック１１６とから構成されている。符号化ブロック１１０に

は、インターリーブされた波長多重通信により送信された入力ディジタル信号が入力され、例えば、リード・ソロモン符号化され、入力ブロックへと符号化されたディジタル信号を入力している。入力ブロック１１２は入力ディジタル信号から順序回路を使用してシンドロームを算出し、その出力を処理ブロック１１４へと送っている。

【０１３５】

処理ブロック１１４には、本発明のアルゴリズムを実行する復号機能が含まれていて、誤り位置と誤り値とを算出する。算出された誤り位置と、誤り値は、出力ブロック１１６へと出力され、誤りが訂正され、出力ディジタル信号として出力を行っている。上述した誤り訂正回路は、複数のハードウエアからなる誤り訂正装置として構成することができる他、半導体技術を利用し、各機能ブロックをシリコン・ウエハ上に構成したＡＳＩＣといった半導体デバイスとして構成することができることはいうまでもないことである。さらには、本発明のアルゴリズムは、誤り訂正装置のファームウエアとして実装することもできるし、またフロッピーディスク、ハードディスク、光ディスク、光磁気ディスクといった記憶媒体に記録されたコンピュータ可読なプログラムとすることもできる。また、本発明のプログラムは、例えばオブジェクト指向のいかなる言語や、例えばＣ言語といったプログラミング言語により、記述し、上述した記録媒体内に保持させて使用することができる。

【０１３６】

上述したように、本発明によれば、高速の光通信分野において特に効果的に誤りを訂正することを可能とする組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提供することができる。

【図面の簡単な説明】

【図１】

従来の復号回路を示した図。

【図２】

従来の光通信用誤り訂正回路を示した図。

【図３】

従来の回路規模と、データ転送速度とをプロットした図。

【図４】

さらに別の従来の復号回路を示した図。

【図５】

さらに別の従来の復号回路を示した図。

【図６】

本発明の復号回路の実施の形態の概略図。

【図７】

従来の構成の乗算回路を示した図。

【図８】

従来の構成の乗算回路を示した図。

【図９】

従来の構成の乗算回路を示した図。

【図１０】

図７に示す乗算回路に本発明を適用した実施の形態を示した図。

【図１１】

図８に示す乗算回路に本発明を適用した実施の形態を示した図。

【図１２】

図９に示す乗算回路に本発明を適用した実施の形態を示した図。

【図１３】

本発明の乗算回路を使用した場合の回路サイズと乗算回路数とをプロットした図。

【図１４】

従来の誤り多項式を示した図。

【図１５】

本発明におけるYule-Walker方程式の定式化を示した図。

【図１６】

本発明における$\Gamma_i^{(i+1)}$の詳細な構成を示した図。

【図１７】

本発明におけるリード・ソロモン符号の復号の詳細な計算結果を示した図。

【図１８】

本発明の誤り訂正アルゴリズムの概略フローチャート。

【図１９】

本発明のリード・ソロモン符号の復号回路の概略構成を示した図。

【図２０】

本発明の誤り訂正装置の構成を示す概略ブロック図。

【符号の説明】

１０…入力部

１２…処理部

１４…出力部

１６…シンドローム算出部

１８…位置多項式係数算出部

２０…誤り値多項式係数算出部

２２…レジスタ

２４…ＡＮＤゲート

２６…ＸＯＲゲート

２８ａ，２８ｂ…ｉｍｎバッファ

４０…乗算回路

４２…乗算回路

４５ａ，４５ｂ…出力

４６…乗算器群

４７…加算器群

５２…ＸＯＲ群

５４…ＸＯＲ群

５６…ＸＯＲ群

６０…下流側ＸＯＲ群

６０ａ，６０ｂ…ＸＯＲ群

６２…出力

６４…出力

６６…入力

６８、７０、７２、７７、７８…入力

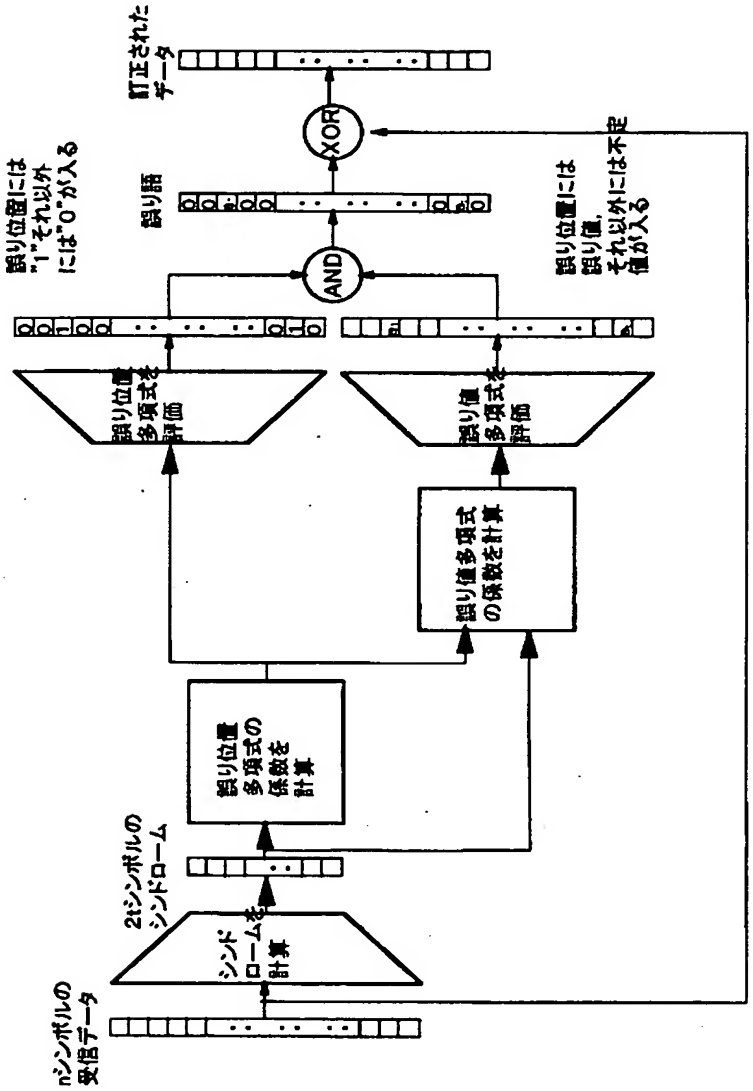
８０…ＡＮＤ群

８２…ＡＮＤ部

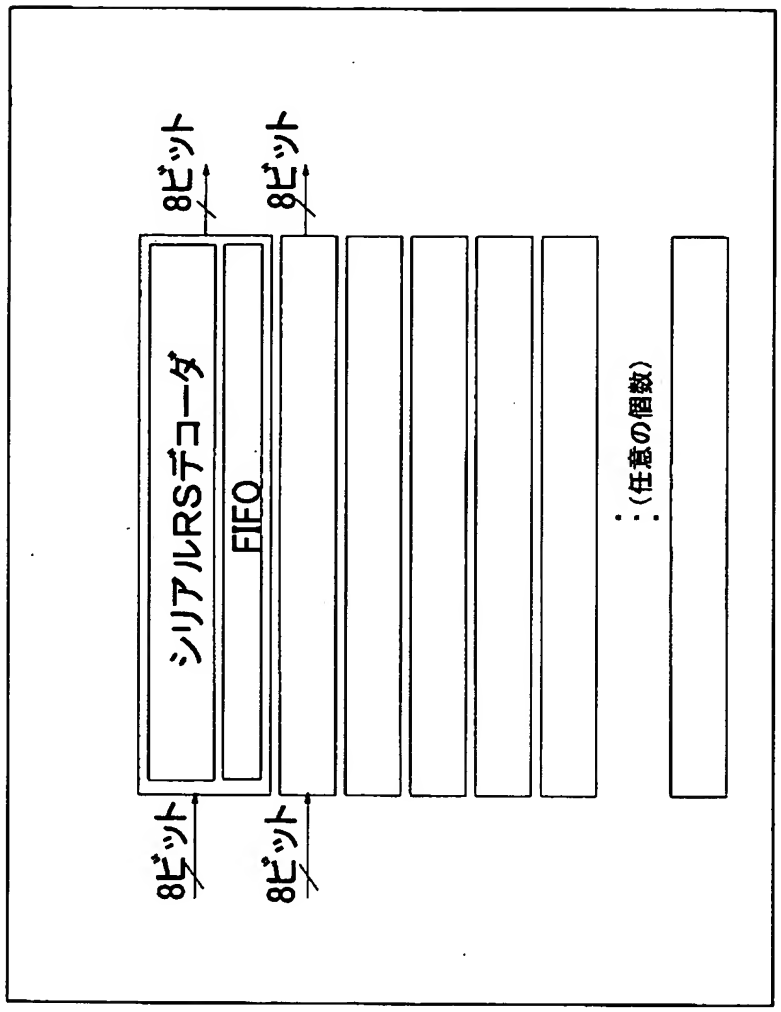
８４…加算器

【書類名】　図面

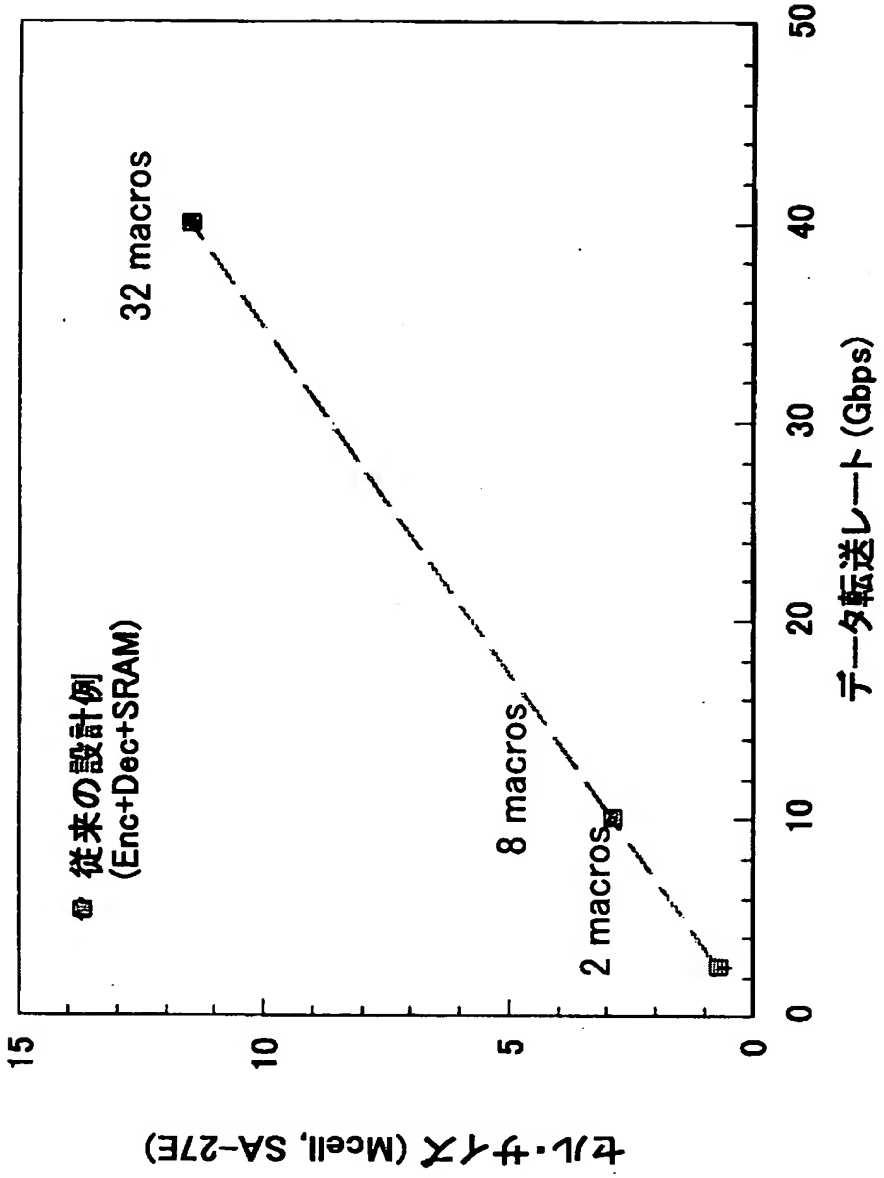
【図１】


訂正されたデータ
XOR
誤り語
AND
誤り位置には"1"それ以外には"0"が入る
誤り位置には誤り値，それ以外には不定値が入る
誤り位置多項式を評価
誤り値多項式を評価
誤り値多項式の係数を計算
誤り位置多項式の係数を計算
2tシンボルのシンドローム
シンドロームを計算
nシンボルの受信データ

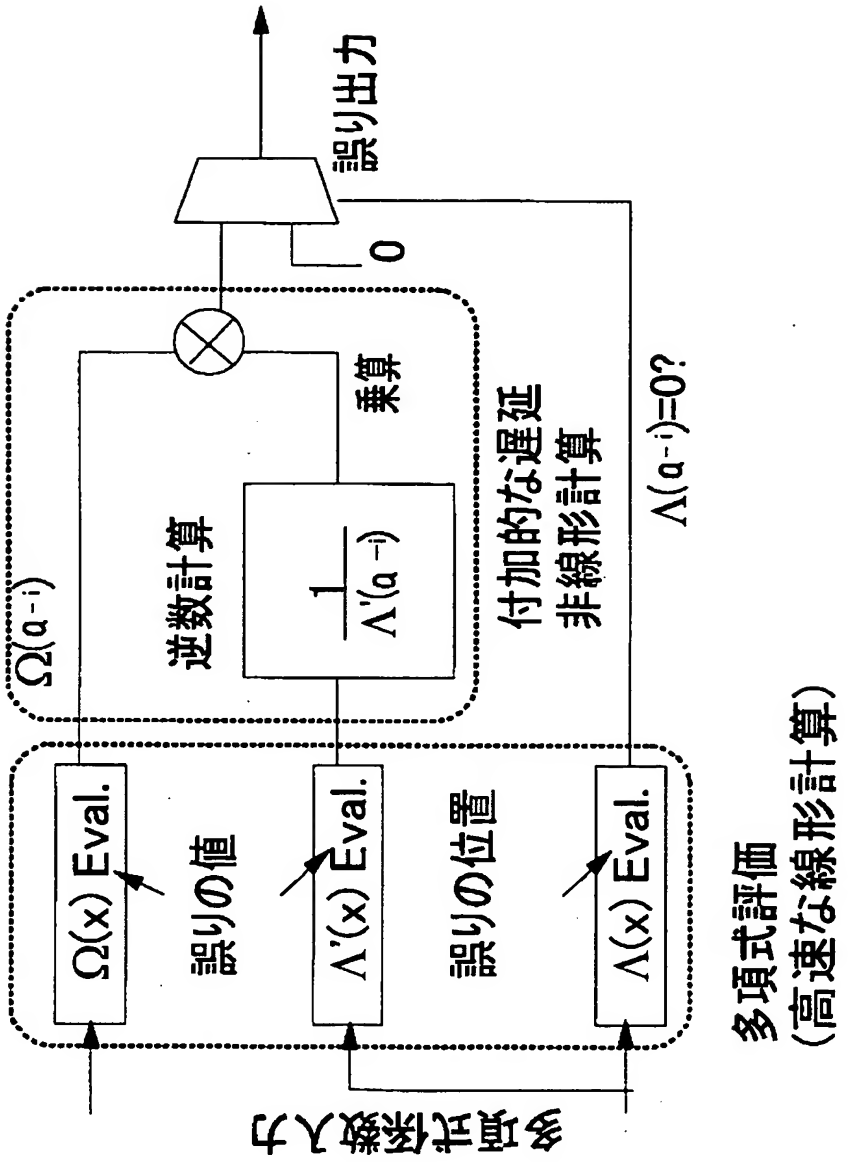
【図２】



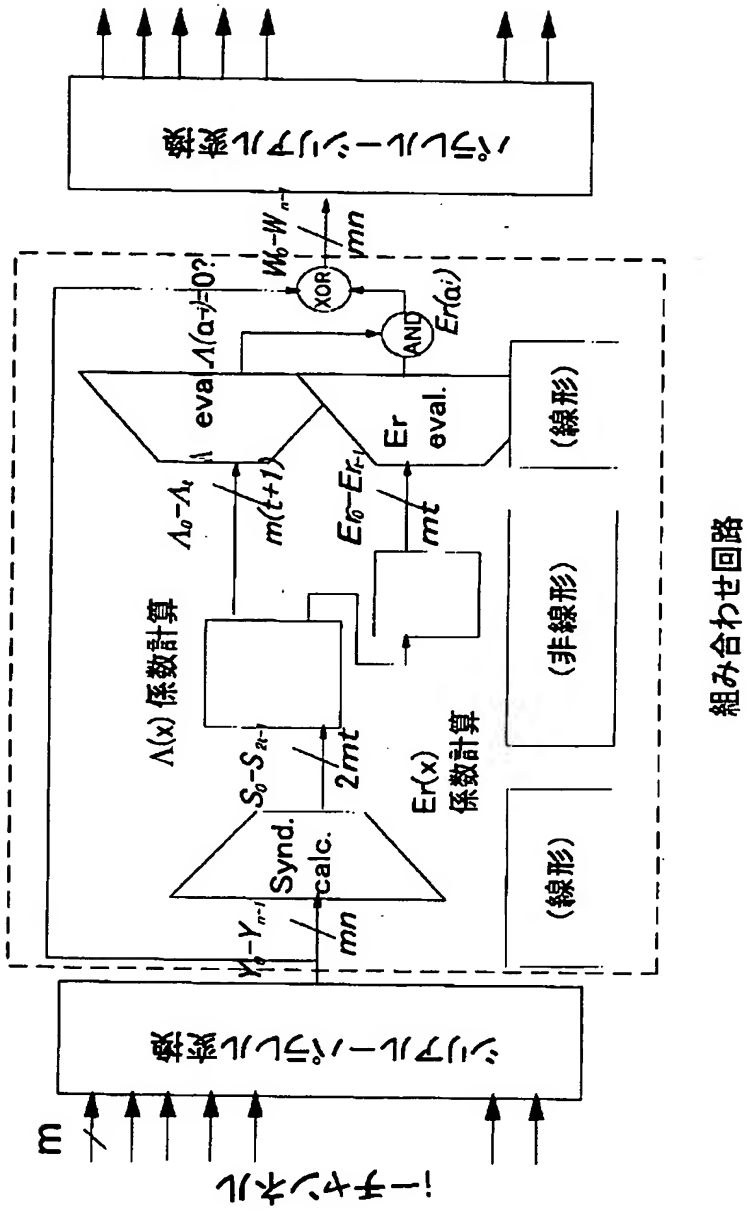
【図３】


32 macros
8 macros
2 macros
従来の設計例
(Enc+Dec+SRAM)
データ転送レート (Gbps)
セル・サイズ (Mcell, SA-27E)
0
10
20
30
40
50
0
5
10
15

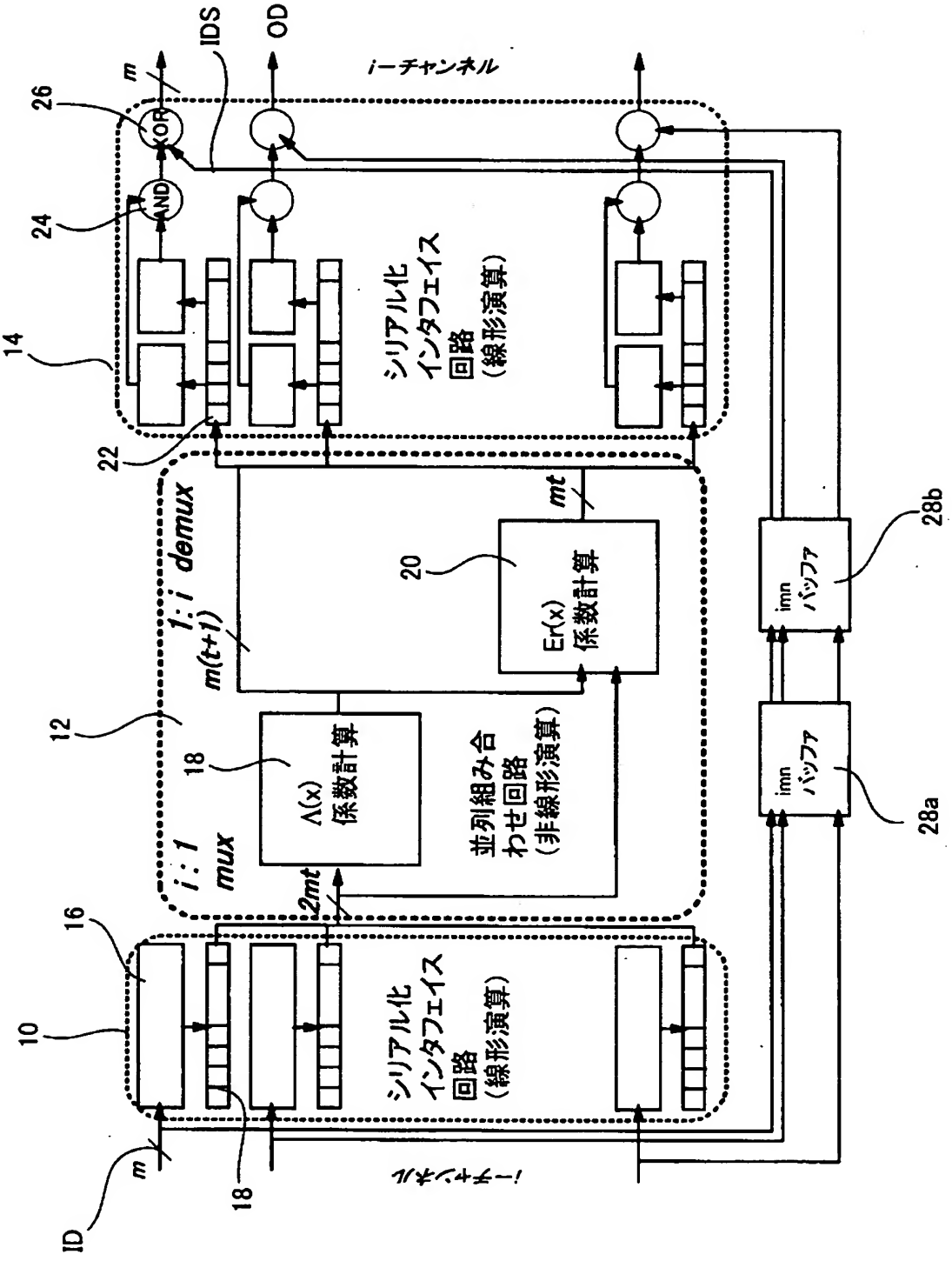
【図４】


誤り出力
0
乗算
逆数計算
付加的な遅延
非線形計算
Λ(α^-i)=0?
Ω(α^-i)
1/Λ'(α^-i)
Ω(x) Eval.
誤りの値
Λ'(x) Eval.
誤りの位置
Λ(x) Eval.
多項式評価
(高速な線形計算)
多項式係数入力

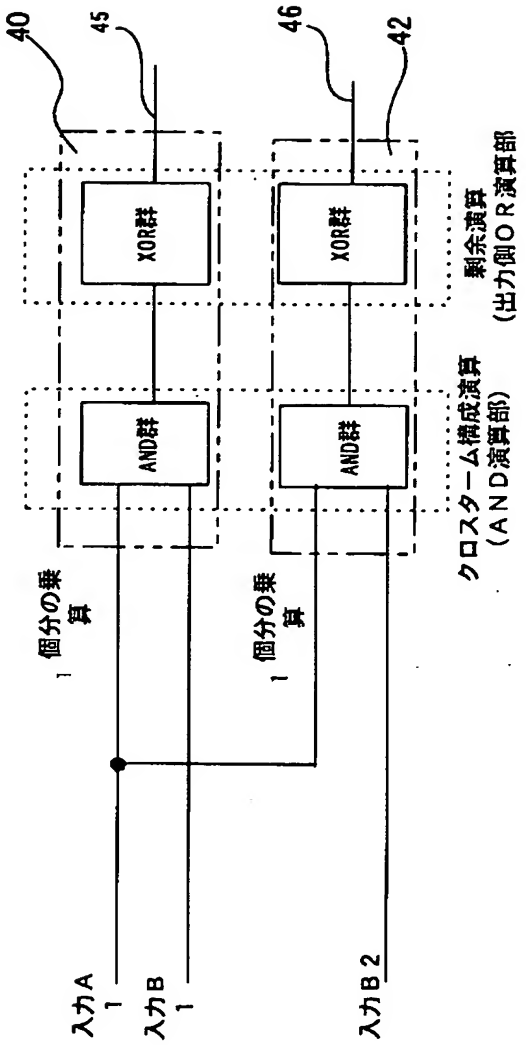
【図５】


ベクトル－インデクス変換
W_0-W_{n-1}
mn
XOR
AND
$Er(a_i)$
$\Lambda(a_j)$=0?
eval.
Er eval.
(線形)
Λ_0-Λ_t
m(t+1)
E_0-E_{r-1}
mt
(非線形)
Λ(x) 係数計算
Er(x) 係数計算
S_0-S_{2t-1}
2mt
Synd. calc.
(線形)
Y_0-Y_{n-1}
mn
インデクス－ベクトル変換
m
シンドローム
組み合わせ回路

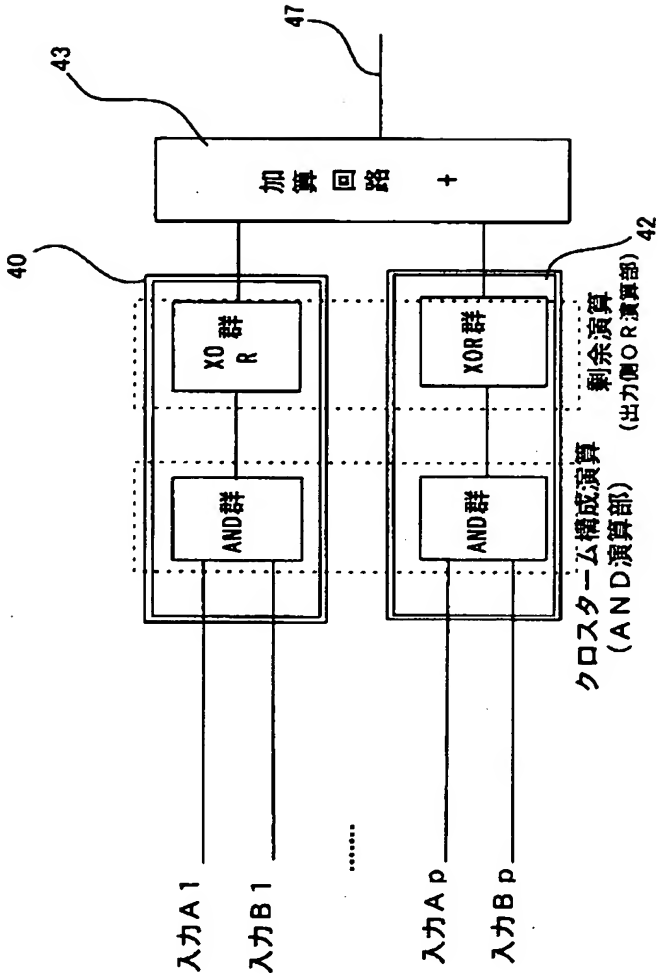
【図６】


IDS
OD
i-チャンネル
m
26
OR
AND
24
14
22
シリアル化
インタフェイス
回路
(線形演算)
1:i demux
m(t+1)
12
mt
20
Er(x)
係数計算
28b
imn
バッファ
18
Λ(x)
係数計算
並列組み合
わせ回路
(非線形演算)
i:1
mux
2mt
28a
16
10
シリアル化
インタフェイス
回路
(線形演算)
m
18
ID
i-チャンネル

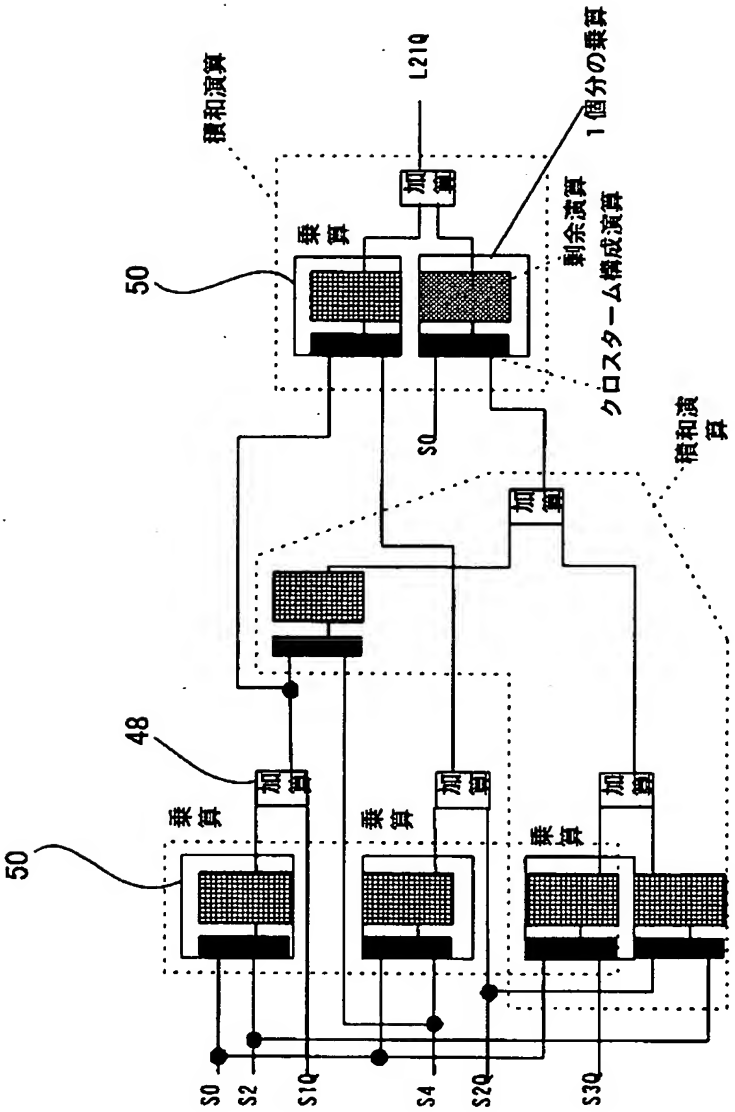
【図７】


40
45
46
42
XOR群
XOR群
AND群
AND群
剰余演算
(出力側ＯＲ演算部)
クロスターム構成演算
(ＡＮＤ演算部)
個分の乗算
1
個分の乗算
1
入力A
1
入力B
1
入力B２

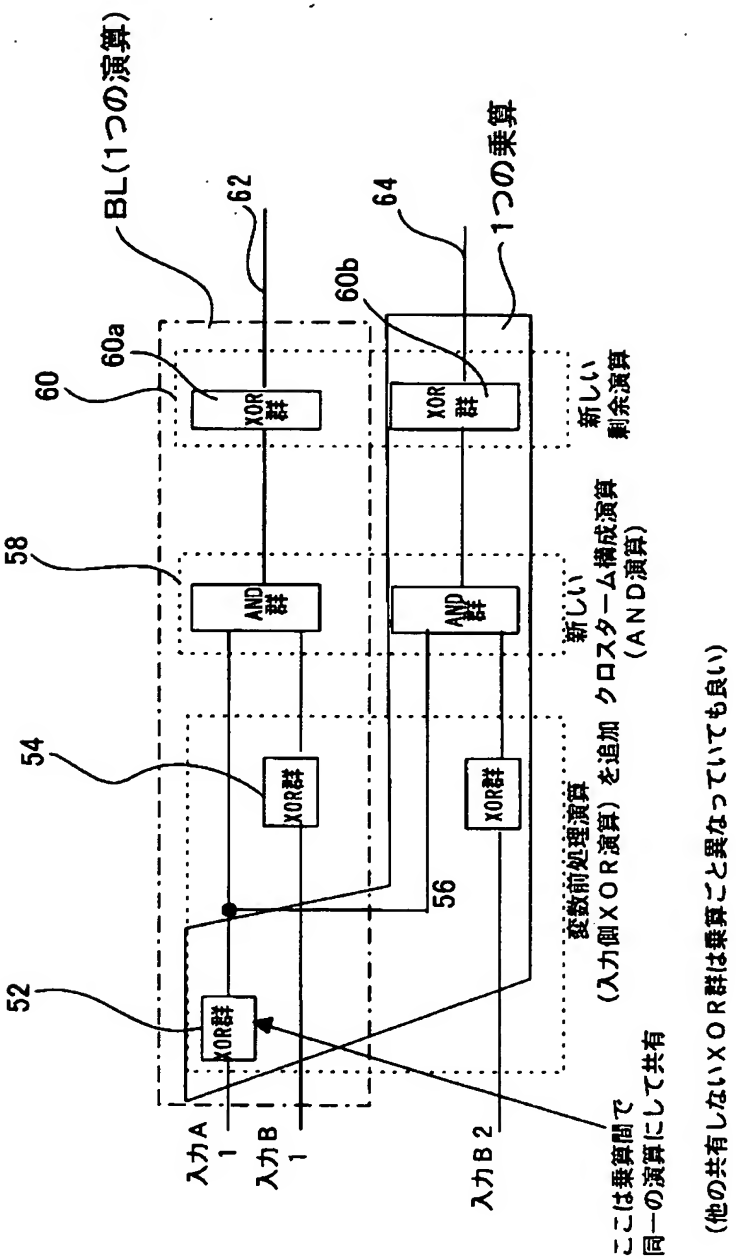
【図８】


47
43
加算回路 +
40
42
XOR群
AND群
XOR群
AND群
剰余演算
(出力側ＯＲ演算部)
クロスターム構成演算
（ＡＮＤ演算部）
入力Ａ１
入力Ｂ１
.....
入力Ａｐ
入力Ｂｐ

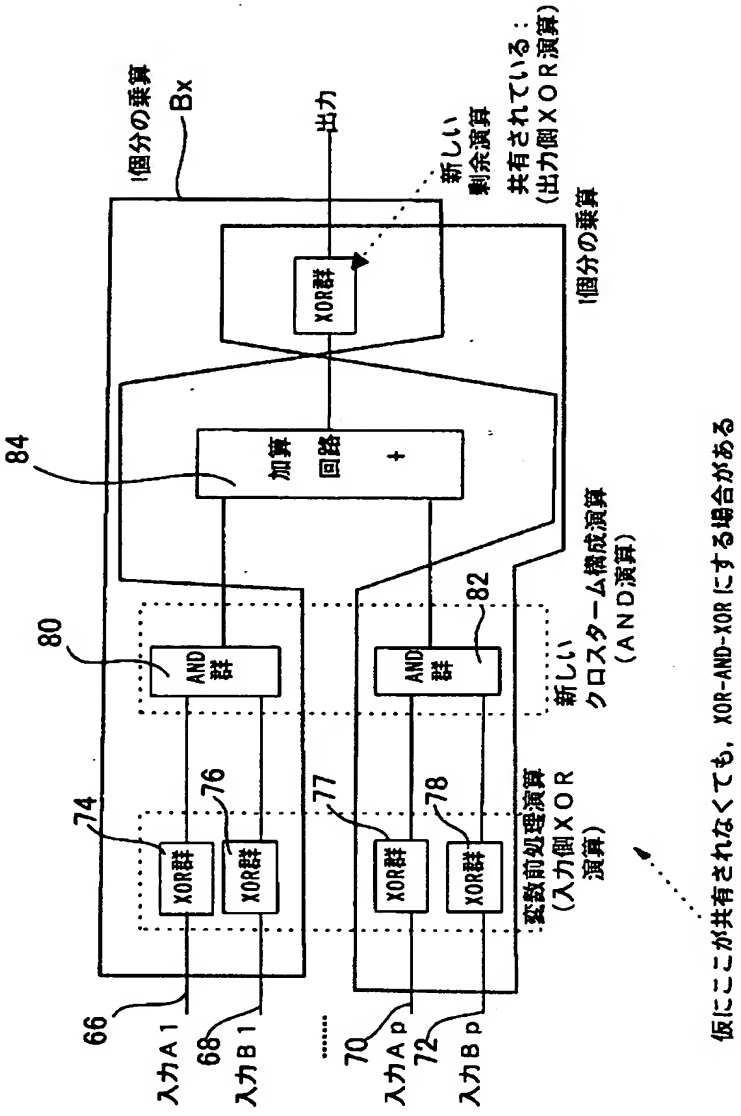
【図９】


積和演算
L21Q
1個分の乗算
加算
乗算
剰余演算
クロスターム構成演算
50
S0
積和演算
48
S0
S2
S10
S4
S20
S30

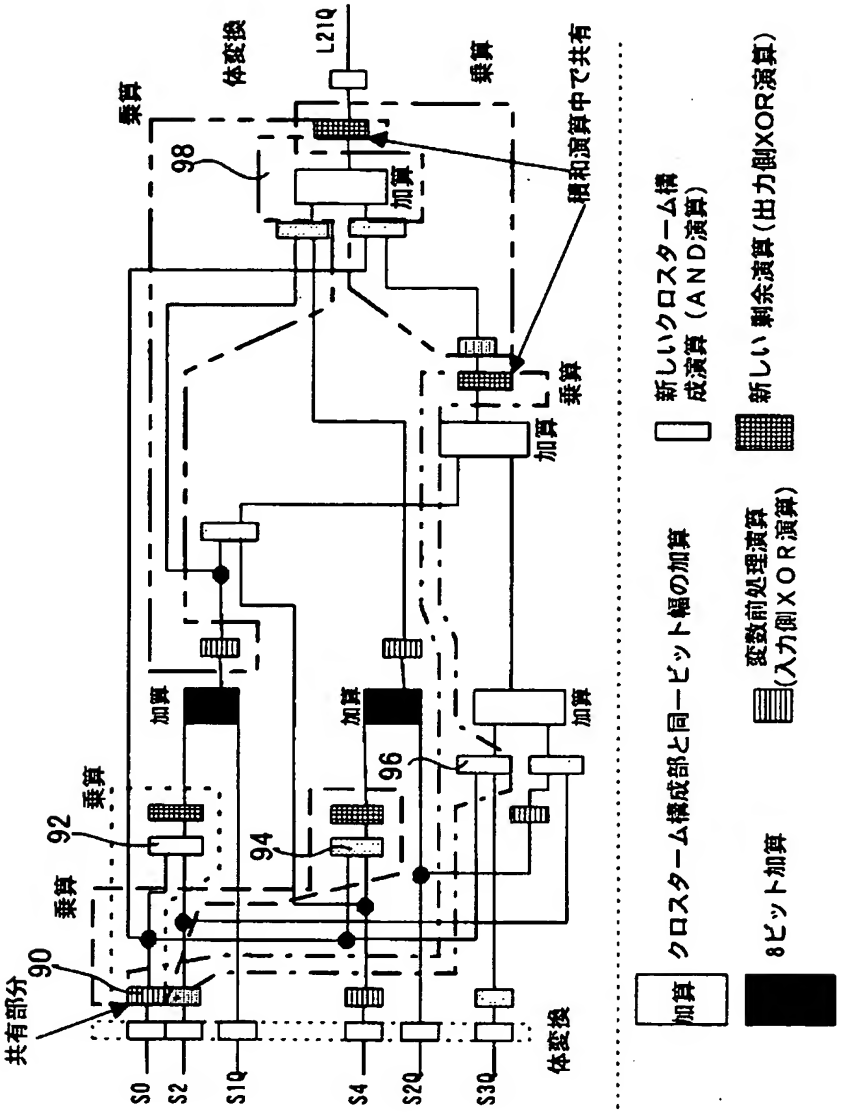
【図１０】


BL(1つの演算)
62
64
1つの乗算
60b
60a
60
XOR群
新しい剰余演算
58
AND群
新しい
クロスターム構成演算
(ＡＮＤ演算)
54
XOR群
変数前処理演算
(入力側ＸＯＲ演算)を追加
56
52
XOR群
入力A 1
入力B 1
入力B 2
ここは乗算間で同一の演算にして共有
(他の共有しないＸＯＲ群は乗算ごと異なっていても良い)

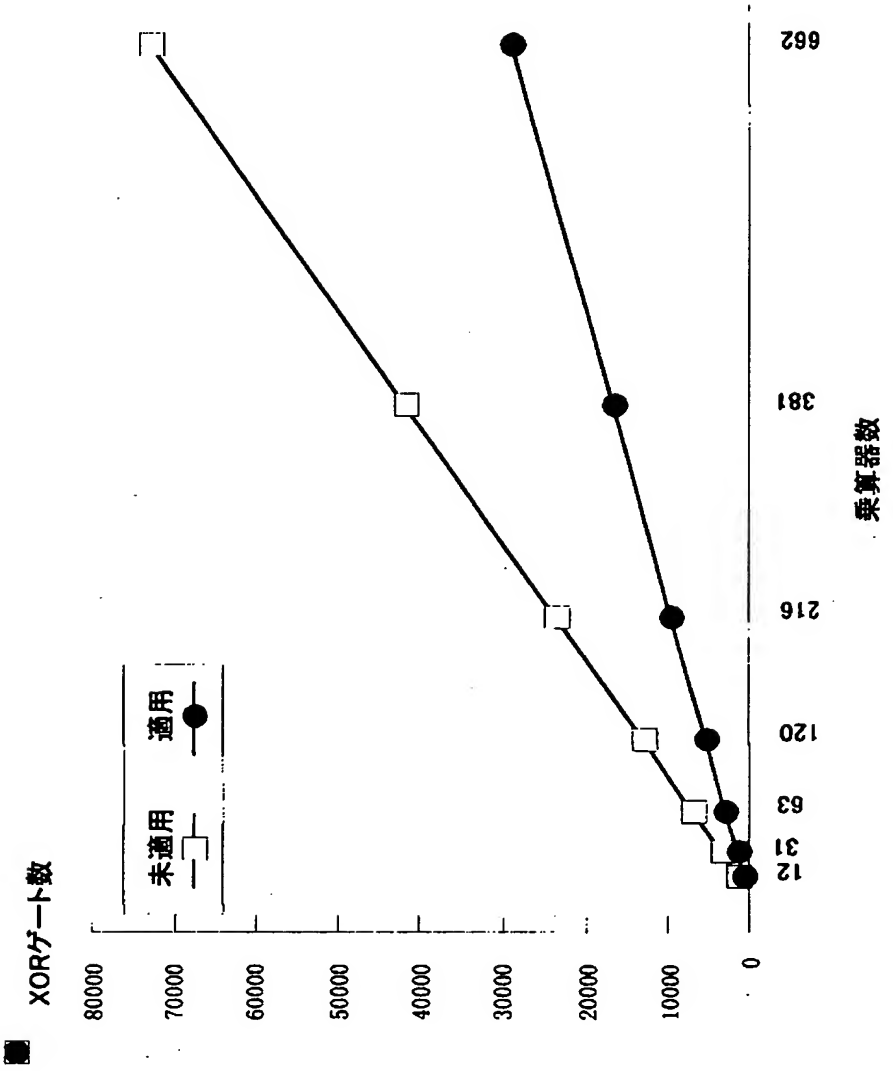
【図１１】


1個分の乗算
Bx
出力
新しい
剰余演算
共有されている：
(出力側ＸＯＲ演算)
1個分の乗算
XOR群
84
加算
回路
+
80
AND
群
82
新しい
クロスターム構成演算
(ＡＮＤ演算)
74
76
77
78
XOR群
XOR群
XOR群
XOR群
変数前処理演算
(入力側ＸＯＲ
演算)
66
入力Ａ１
68
入力Ｂ１
.....
70
入力Ａｐ
72
入力Ｂｐ
仮にここが共有されなくても，XOR-AND-XORにする場合がある

【図１２】


乗算
体変換
L21Q
乗算
98
加算
積和演算中で共有
乗算
加算
加算
加算
加算
96
乗算
92
94
乗算
90
共有部分
S0
S2
S1Q
S4
S2Q
S3Q
体変換
新しいクロスターム構成演算（ＡＮＤ演算）
新しい剰余演算（出力側XOR演算）
クロスターム構成部と同一ビット幅の加算
変数前処理演算（入力側ＸＯＲ演算）
加算
8ビット加算

【図１３】


662
381
216
120
63
31
12
0
乗算器数
適用
未適用
80000
70000
60000
50000
40000
30000
20000
10000
XORゲート数

【図１４】

$$\tilde{\Lambda}_1^{(1)} = S_1$$
$$\tilde{\Lambda}_0^{(1)} = S_0$$

$l = 2$の場合

$$\tilde{\Lambda}_2^{(2)} = \tilde{\Lambda}_1^{(1)} S_3 + S_2^2$$
$$\tilde{\Lambda}_1^{(2)} = \tilde{\Lambda}_0^{(1)} S_3 + \tilde{\Lambda}_1^{(1)} S_2$$
$$\tilde{\Lambda}_0^{(2)} = \tilde{\Lambda}_0^{(1)} S_2 + \tilde{\Lambda}_1^{(1)} S_1$$

$l = 3$の場合

$$\tilde{\Lambda}_3^{(3)} = \tilde{\Lambda}_2^{(2)} S_5 + S_1 S_4^2 + S_3^3$$
$$\tilde{\Lambda}_2^{(3)} = \tilde{\Lambda}_1^{(2)} S_5 + \tilde{\Lambda}_2^{(2)} S_4 + S_0 S_4^2 + S_2 S_3^2$$
$$\tilde{\Lambda}_1^{(3)} = \tilde{\Lambda}_0^{(2)} S_5 + \tilde{\Lambda}_1^{(2)} S_4 + \tilde{\Lambda}_2^{(2)} S_3$$
$$\tilde{\Lambda}_0^{(3)} = \tilde{\Lambda}_0^{(2)} S_4 + \tilde{\Lambda}_1^{(2)} S_3 + \tilde{\Lambda}_2^{(2)} S_2$$

$l = 4$の場合

$$\tilde{\Lambda}_4^{(4)} = \tilde{\Lambda}_3^{(3)} S_7 + \tilde{\Lambda}_2^{(2)} S_6^2 + S_4^4 + S_3 S_4^2 S_5 + S_3^2 S_5^2 + S_1 S_5^3$$
$$\tilde{\Lambda}_3^{(4)} = \tilde{\Lambda}_2^{(3)} S_7 + \tilde{\Lambda}_3^{(3)} S_6 + \tilde{\Lambda}_1^{(2)} S_6^2 + S_3 S_4^3 + S_2 S_4^2 S_5 + S_1 S_4 S_5^2 + S_0 S_5^3$$
$$\tilde{\Lambda}_2^{(4)} = \tilde{\Lambda}_1^{(3)} S_7 + \tilde{\Lambda}_2^{(3)} S_6 + \tilde{\Lambda}_3^{(3)} S_5 + \tilde{\Lambda}_0^{(2)} S_6^2 + S_3^2 S_4^2 + S_2 S_4^3 + S_2^2 S_5^2 + S_0 S_4 S_5^2$$
$$\tilde{\Lambda}_1^{(4)} = \tilde{\Lambda}_0^{(3)} S_7 + \tilde{\Lambda}_1^{(3)} S_6 + \tilde{\Lambda}_2^{(3)} S_5 + \tilde{\Lambda}_3^{(3)} S_4$$
$$\tilde{\Lambda}_0^{(4)} = \tilde{\Lambda}_0^{(3)} S_6 + \tilde{\Lambda}_1^{(3)} S_5 + \tilde{\Lambda}_2^{(3)} S_4 + \tilde{\Lambda}_3^{(3)} S_3.$$

【図１５】

$$\Lambda_i^{(l)} = \frac{\tilde{\Lambda}_i^{(l)}}{\tilde{\Lambda}_0^{(l)}}, \quad i = 1, \cdots, l$$

$$\tilde{\Lambda}_0^{(l)} = \begin{vmatrix} S_0 & \cdots & S_{l-1} \\ \vdots & \ddots & \vdots \\ S_{l-1} & \cdots & S_{2l-2} \end{vmatrix},$$

$$\tilde{\Lambda}_i^{(l)} = \begin{vmatrix} S_0 & \cdots & S_{l-1} \\ \vdots & \ddots & \vdots \\ S_{l-i-1} & \cdots & S_{2l-i-2} \\ S_{l-i+1} & \cdots & S_{2l-i} \\ \vdots & \ddots & \vdots \\ S_l & \cdots & S_{2l-1} \end{vmatrix}, \quad i = 1, \cdots, l-1$$

$$\tilde{\Lambda}_l^{(l)} = \begin{vmatrix} S_1 & \cdots & S_l \\ \vdots & \ddots & \vdots \\ S_l & \cdots & S_{2l-1} \end{vmatrix}.$$

【図１６】

$$\Gamma_0^{(l+1)} = \begin{vmatrix} S_0 & S_1 & \cdots & S_{l-1} \\ S_1 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ S_{l-1} & \cdots & \cdots & S_{2l-2} \end{vmatrix},$$

$$\Gamma_i^{(l+1)} = \begin{vmatrix} S_0 & \cdots & S_{l-1-i} & S_{l+1-i} & \cdots & S_l \\ \vdots & & \vdots & \vdots & & \vdots \\ S_{l-1-i} & \cdots & S_{2(l-1-i)} & S_{2(l-i)} & \cdots & S_{2l-1-i} \\ S_{l+1-i} & \cdots & S_{2(l-i)} & S_{2(l+1-i)} & \cdots & S_{2l+1-i} \\ \vdots & & \vdots & \vdots & & \vdots \\ S_l & \cdots & S_{2l-1-i} & S_{2l+1-i} & \cdots & S_{2l} \end{vmatrix} \qquad i = 1, \cdots, l-1$$

$$\Gamma_l^{(l+1)} = \begin{vmatrix} S_2 & \cdots & \cdots & S_{l+1} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ S_{l+1} & \cdots & \cdots & S_{2l} \end{vmatrix}.$$

【図１７】

$$\Gamma_0^{(1)} = 1$$

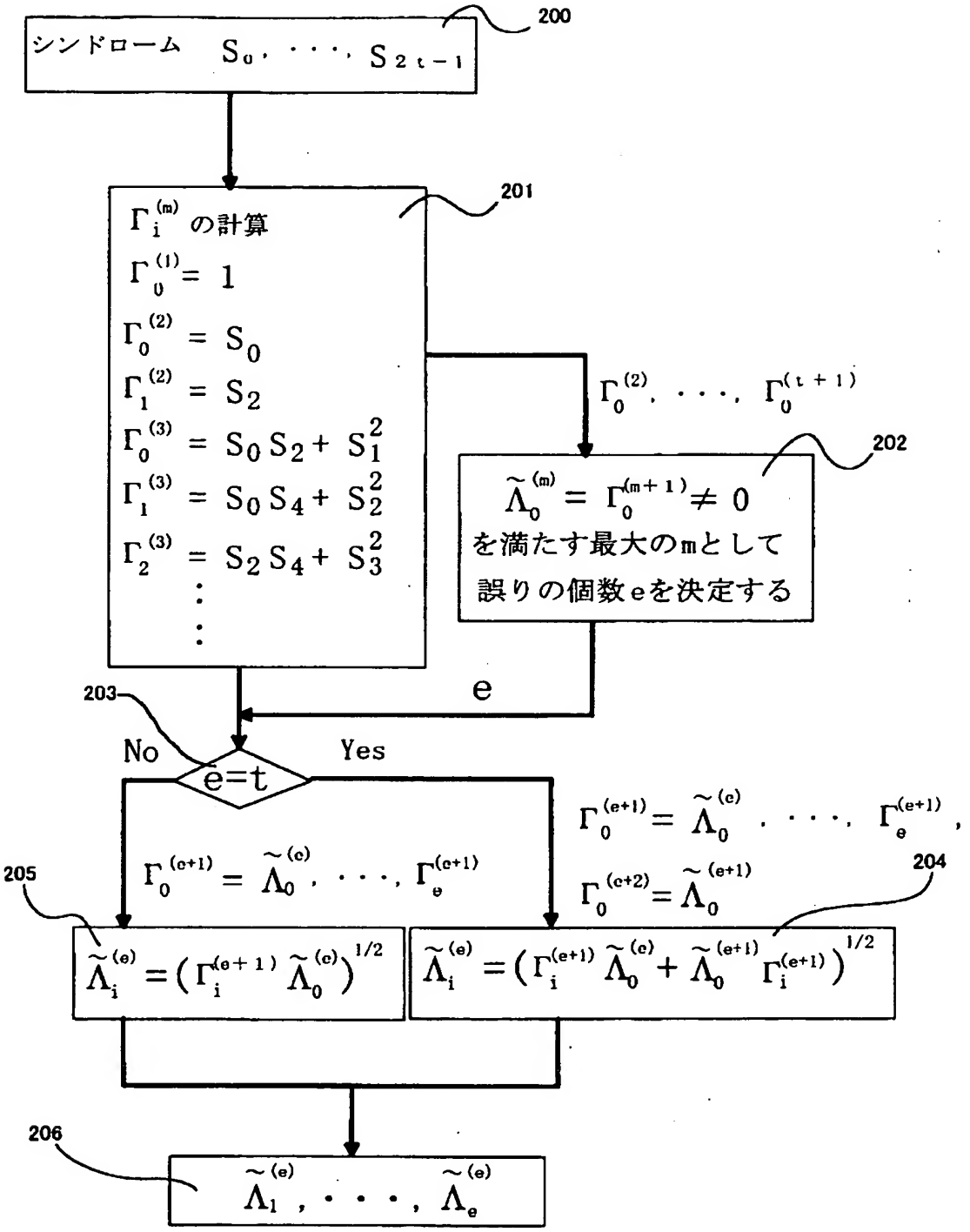
$$\Gamma_0^{(2)} = S_0$$
$$\Gamma_1^{(2)} = S_2$$

$$\Gamma_0^{(3)} = S_0S_2 + S_1^2$$
$$\Gamma_1^{(3)} = S_0S_4 + S_2^2$$
$$\Gamma_2^{(3)} = S_2S_4 + S_3^2$$

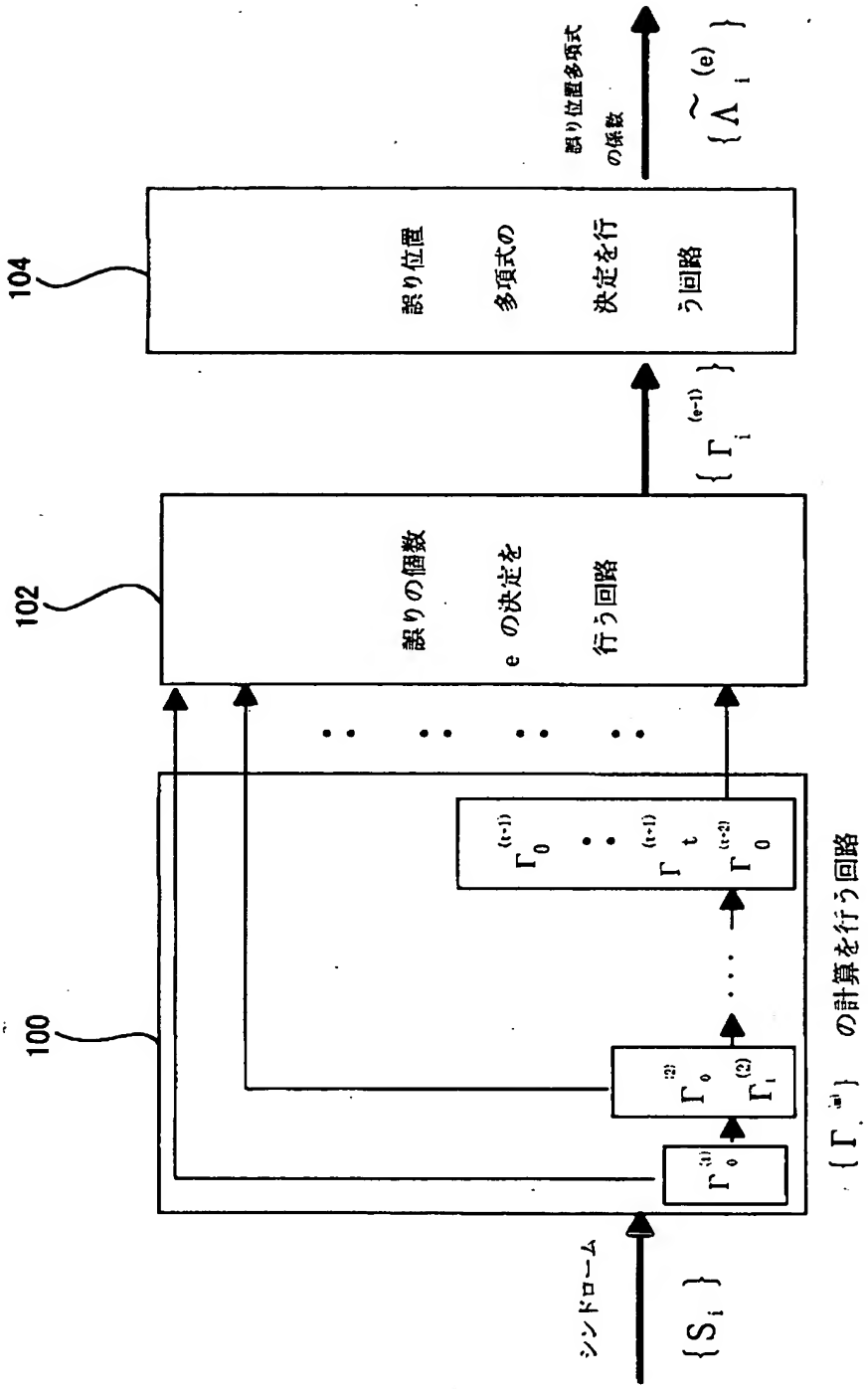
$$\Gamma_0^{(4)} = \Gamma_0^{(3)}S_4 + S_0S_3^2 + S_2^3$$
$$\Gamma_1^{(4)} = \Gamma_0^{(3)}S_6 + S_0S_4^2 + S_2S_3^2$$
$$\Gamma_2^{(4)} = \Gamma_1^{(3)}S_6 + S_0S_5^2 + S_4S_3^2$$
$$\Gamma_3^{(4)} = \Gamma_2^{(3)}S_6 + S_2S_5^2 + S_4S_4^2$$

$$\Gamma_0^{(5)} = \Gamma_0^{(4)}S_6 + \Gamma_0^{(3)}S_5^2 + \Gamma_1^{(3)}S_4^2 + \Gamma_2^{(3)}S_3^2$$
$$\Gamma_1^{(5)} = \Gamma_0^{(4)}S_8 + \Gamma_0^{(3)}S_6^2 + \Gamma_1^{(3)}S_5^2 + \Gamma_2^{(3)}S_4^2$$
$$\det 03 = S_0S_6 + S_3^2$$
$$\det 24 = S_2S_6 + S_4^2$$
$$\Gamma_2^{(5)} = \Gamma_1^{(4)}S_8 + \Gamma_0^{(3)}S_7^2 + \det 03 \cdot S_5^2 + \det 24 \cdot S_4^2$$
$$\det 45 = S_4S_6 + S_5^2$$
$$\Gamma_3^{(5)} = \Gamma_2^{(4)}S_8 + \Gamma_1^{(3)}S_7^2 + \det 03 \cdot S_6^2 + \det 45 \cdot S_4^2$$
$$\Gamma_4^{(5)} = \Gamma_3^{(4)}S_8 + \Gamma_2^{(3)}S_7^2 + \det 24 \cdot S_8^2 + \det 45 \cdot S_5^2$$
$$\Gamma_0^{(6)} = \Gamma_0^{(5)}S_8 + \Gamma_0^{(4)}S_7^2 + \Gamma_1^{(4)}S_6^2 + \Gamma_2^{(4)}S_5^2 + \Gamma_3^{(4)}S_4^2.$$

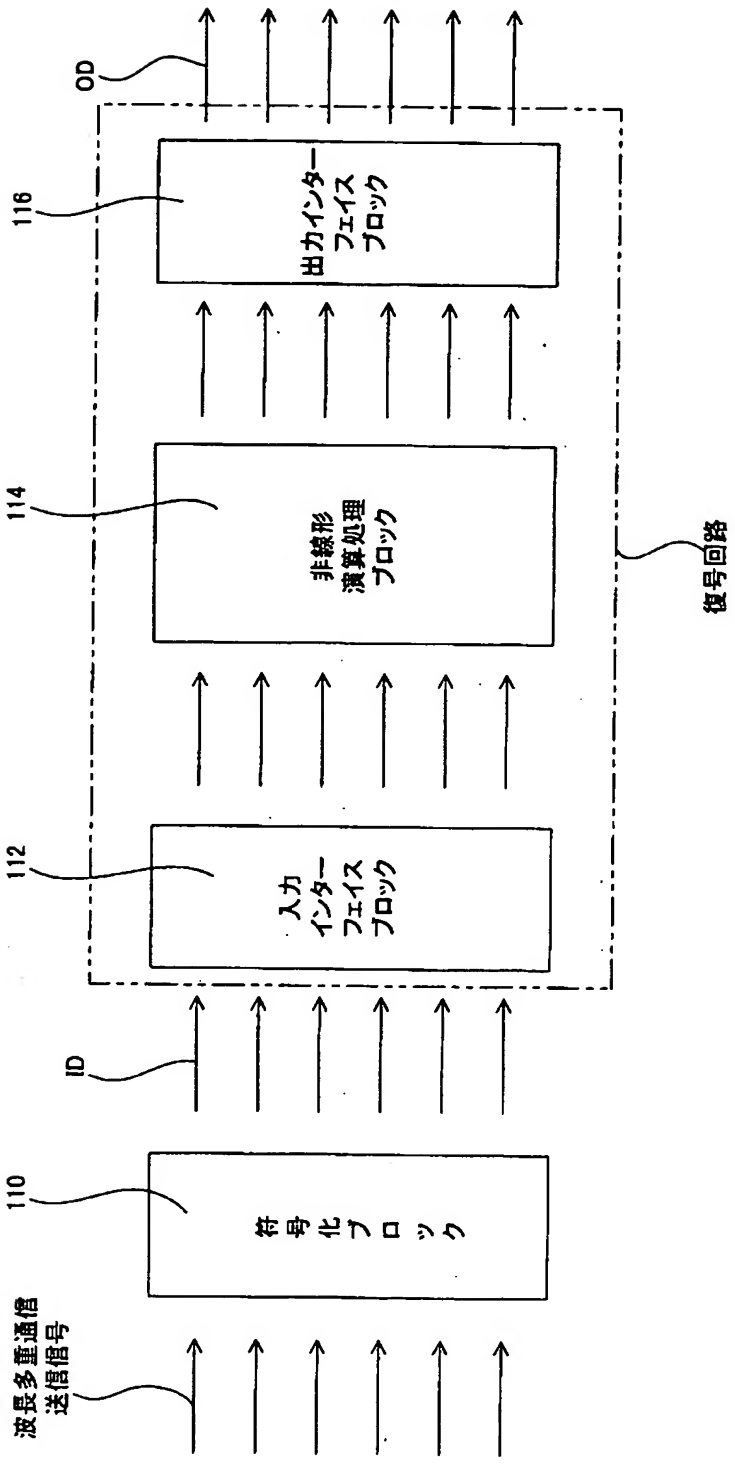
【図１８】


200
シンドローム $S_0, \cdots, S_{2t-1}$
201
$\Gamma_i^{(m)}$ の計算
$\Gamma_0^{(1)} = 1$
$\Gamma_0^{(2)} = S_0$
$\Gamma_1^{(2)} = S_2$
$\Gamma_0^{(3)} = S_0 S_2 + S_1^2$
$\Gamma_1^{(3)} = S_0 S_4 + S_2^2$
$\Gamma_2^{(3)} = S_2 S_4 + S_3^2$
$\Gamma_0^{(2)}, \cdots, \Gamma_0^{(t+1)}$
202
$\tilde{\Lambda}_0^{(m)} = \Gamma_0^{(m+1)} \neq 0$
を満たす最大のmとして
誤りの個数eを決定する
e
203
No
e=t
Yes
$\Gamma_0^{(e+1)} = \tilde{\Lambda}_0^{(e)}, \cdots, \Gamma_e^{(e+1)}$,
$\Gamma_0^{(e+2)} = \tilde{\Lambda}_0^{(e+1)}$
204
$\Gamma_0^{(e+1)} = \tilde{\Lambda}_0^{(e)}, \cdots, \Gamma_e^{(e+1)}$
205
$\tilde{\Lambda}_i^{(e)} = (\Gamma_i^{(e+1)} \tilde{\Lambda}_0^{(e)})^{1/2}$
$\tilde{\Lambda}_i^{(e)} = (\Gamma_i^{(e+1)} \tilde{\Lambda}_0^{(e)} + \tilde{\Lambda}_0^{(e+1)} \Gamma_i^{(e+1)})^{1/2}$
206
$\tilde{\Lambda}_1^{(e)}, \cdots, \tilde{\Lambda}_e^{(e)}$

【図１９】


104
誤り位置多項式の係数
誤り位置多項式の決定を行う回路
102
誤りの個数eの決定を行う回路
100
シンドローム
{Si}
{Γ(e)}の計算を行う回路

【図２０】


OD
116
出力インターフェイスブロック
114
非線形演算処理ブロック
復号回路
112
入力インターフェイスブロック
ID
110
符号化ブロック
波長多重通信送信信号

【書類名】　要約書

【要約】

【課題】　組み合わせ回路、該組み合わせ回路を使用する符号化装置、復号装置、および半導体デバイスを提供する。

【解決手段】　本発明の組み合わせ回路は、ガロア拡大体ＧＦ（2^m）（ｍは、２以上の整数）における符号化されたディジタル信号の２個以上の乗算を独立して行う複数の乗算器を含み、乗算器は、入力側ＸＯＲ演算部と、ＡＮＤ演算部と、出力側ＸＯＲ演算部とを含んで構成され、入力側ＸＯＲ演算部を、複数の乗算器が共有する。また、本発明においては上述した乗算器は、ＡＮＤ演算部と、出力側ＸＯＲ演算部との間に接続される加算器を含み、出力側ＸＯＲ演算部が共有され、複数の前記乗算器のＡＮＤ演算部の出力を加算器により加算し、その加算結果を共有される出力側ＸＯＲ演算部により演算することもできる。

【選択図】　図１０

出　願　人　履　歴　情　報

識別番号　　〔３９０００９５３１〕

１．変更年月日　　２０００年　５月１６日
［変更理由］　名称変更
住　所　　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　（番地なし）
氏　名　　インターナショナル・ビジネス・マシーンズ・コーポレーション